

PROFES



PROFES

USER MANUAL

Final Version 1999



TABLE OF CONTENTS:

1. INTRODUCTION	1-1
2. PROFES IMPROVEMENT METHODOLOGY OVERVIEW	2-1
3. PROFES STEPS	3-1
4. ADVANCED TECHNIQUES	4-1
5. COST/BENEFIT OF PROFES	5-1
6. THE PROFES TOOLS	6-1
7. PRODUCT/PROCESS DEPENDENCIES IN PROFES	7-1
8. BUILDING INFRASTRUCTURE FOR PROFES	8-1
9. REFERENCES	9-1

APPENDICES:

APPENDIX 1: OVERVIEW OF THE PROFES IMPROVEMENT METHODOLOGY ELEMENTS	A1-1
APPENDIX 2: THE PROFES TEMPLATES	A2-1
APPENDIX 3: EXAMPLES OF PRODUCT PROCESS DEPENDENCIES	A3-1
APPENDIX 4: COST/BENEFIT MODELS OF PROFES	A4-1
APPENDIX 5: THE PROFES TOOLS	A5-1
APPENDIX 6: THE PROFES GLOSSARY	A6-1
APPENDIX 7: PROFES QUICK REFERENCE CHART	A7-1



Chapter 1

WHY PROFES IMPROVEMENT METHODOLOGY?

The complexity of software development is continuously increasing, and the need to shorten lead-time becomes more pressing. Much time and effort has been spent on assessing and improving software processes. However, exact knowledge of the effects that specific process improvement actions have on specific customer-defined product quality characteristics has not yet been systematically investigated. Consequently, generic process improvement is no longer sufficient and it is therefore essential to shift emphasis towards focused improvement of software processes, based on explicit product quality requirements. Rapid and cost-effective process improvement needs to be linked to product quality.

PROFES is a methodology that helps to improve product quality by focusing process improvement on those parts of the process that have a direct effect on the customer-oriented product quality requirements of a particular organization.

To the Reader

The purpose of this user manual is to provide a detailed description of the PROFES improvement methodology, intended for use in product-driven process improvement. Together with the PROFES tools, this manual provides comprehensive assistance in applying this methodology. The PROFES improvement methodology is adaptable, so that it can be easily applied in different organizations and different software development environments, and can take advantage of investments already made in process improvement, such as an established CMM assessment culture.

The main target audience for this user manual includes professionals, who are actively involved in software process improvement (SPI). They may be either consultants helping companies in software process improvement, or internal experts in an organization that is interested in SPI. This user manual is also helpful for managers and software practitioners who are interested in acquiring detailed knowledge of product-driven process improvement. However, busy readers interested in the principles of the PROFES improvement methodology may choose to read the more concise PROFES book, which introduces the PROFES concepts and ideas as well as presents experiences from three industrial applications of the PROFES improvement methodology.

The PROFES methodology was developed through the integration of software process assessment, software measurement, and organizational learning through the relationships between product and process characteristics. Therefore, the resulting methodology provides support for:

- Planning and performing process improvement driven by product characteristics and objectives
- Estimating costs of improvement efforts and evaluating them against expected benefits
- Addressing issues of major relevance for executives and managers, such as product quality, delivery precision, and production costs

PROFES in a Nutshell

The PROFES improvement approach is focused on customer-oriented product quality as a driving force for the definition and implementation of process changes. The PROFES improvement methodology integrates software process assessment, product and process modelling, software measurement, and organizational learning. It is supported by operational guidelines and tools to plan and carry out product quality-driven software process improvement. All the background elements are usually applied separately, or are even seen as alternatives to each other, but in PROFES they are used together as a unique methodology. The key element of the PROFES improvement approach is the definition of the relationships and interdependence between process and product quality characteristics. These product/process dependencies are examined in the context of their environment, and summarized as product/process dependency models.

The PROFES project

The PROFES improvement methodology was developed in the PROFES Esprit project between January 1997 and September 1999. The project budget was 3.2 M Euros and EU funding amounted to 1.7 M Euros. The

total effort was 323 person-months, and the project consortium consisted of highly skilled methodology providers and practitioners with comprehensive expertise in process improvement:

Table 1.1. The PROFES Consortium

Partner	Role
Dräger, The Netherlands	<i>Application provider</i>
Ericsson, Finland	<i>Application provider</i>
Etnoteam S.P.A., Italy	<i>Method provider</i>
Fraunhofer IESE, Germany	<i>Method provider</i>
Tokheim, The Netherlands	<i>Application provider</i>
University of Oulu, Finland	<i>Method provider</i>
VTT Electronics, Finland,	<i>Project leader and method provider</i>

One principle of the PROFES project was to reuse results from previous European projects. This applied both to methodological development and tool development. The PROFES project had an excellent opportunity to reuse the previous results from earlier international R&D projects, since the consortium included partners who had been key developers of the underlying methods and methodologies, i.e. BOOTSTRAP assessment, Goal-Question-Metrics, Experience Factory, and the Quality Improvement Paradigm.

The results of the PROFES project help to improve European competitiveness in four main ways:

1. By improving the effectiveness of product development in the participating companies
2. By exploiting the results via the PROFES interest group (www.profes.org)
3. By providing product and process improvement consulting services by Etnoteam, VTT, and Fraunhofer IESE
4. By disseminating information at exhibitions, fairs, tutorials, in a user manual, PROFES book, conference presentations and journal articles to a wider audience.

The PROFES Improvement Methodology

The PROFES improvement methodology is a modular approach, providing support and assistance in the following areas:

- Characterization and assessment of the product quality requirements
- Characterization and assessment of software development processes
- Building, packaging, and application of product/process dependency (PPD) models for product quality-driven process improvement.
- Defining and implementing product quality improvement through process improvement
- Establishing a company-wide measurement programme for improvement monitoring

The PROFES improvement methodology combines and enhances the strengths of well known and widely used methods, such as goal-oriented measurement, process assessment, product and process modelling, and Experience factory. The goal-oriented measurement methodology GQM (Goal/Question/Metric) and the ISO 15504-compliant process assessment and improvement methodology provide the framework used for characterizing and assessing software processes. The ISO 9126 standard is used as a background reference for product quality characteristics. Process modelling is necessary to describe software development processes.

Furthermore, PROFES introduces a new method for establishing product process dependencies (PPD), a unique approach to software process improvement. The initial PPD models have been built in three industrial organizations, which offered real-life experimental environments for methodology development and validation.

Components of the PROFES improvement methodology kit include:

- A PROFES book that introduces the methodology, describes its background and design rationales, and shows examples and motives for its use.
- A PROFES User Manual, which is an operational guide for industrial users for integrated application of software process assessment, process modelling, PPD modelling, goal-oriented measurement, and Experience factory to improve final product quality.
- PROFES tools consisting of existing (pre-)commercial tools integrated and enhanced with additional functions to provide support for integrated

use of on-line process assessment, goal oriented measurement and improvement monitoring.

- Training materials consisting of a professional course, including handouts, a course agenda, and recommended lectures.
- Presentation material and tutorials which explain the benefits deriving from PROFES adoption and support the dissemination and exploitation activities.

Much of the PROFES material is freely available in the PROFES web site (www.profes.org). There is also a PROFES Interest Group that promotes the use of the PROFES methodology. Joining instructions can be found from the PROFES web site.

Structure of the PROFES User Manual

Chapter 2 explains in practical terms *how to proceed* in product quality-driven process improvement, giving an overview of the PROFES improvement methodology. Chapter 2 contains also guidance for tailoring the PROFES improvement methodology.

Chapter 3 describes the basic *PROFES steps and activities* in detail. In addition, the input and output work products, methods, techniques, tools, and templates are also included. An estimate of the required effort and average duration of each step is also given.

Chapter 4 outlines advanced assessment and measurement techniques that can be used to obtain additional benefits from the PROFES improvement methodology.

Chapter 5 presents *cost and benefits* associated with the application of the PROFES improvement methodology.

Chapter 6 discusses issues with establishing *tool support* for the product quality-driven process improvement.

Chapter 7 describes the principles and possibilities of using *product/process dependencies* in the product quality-driven process improvement.

Chapter 8 provides information on *building the infrastructure* for product quality-driven process improvement.

References to existing literature and documentation can be found in Chapter 9.

The *Appendix* section presents an overview of methodology elements that provide a basis for the PROFES improvement methodology, and templates that can be used in product-based process improvement. The appendix also gives examples of product process dependency and cost models. Finally, a glossary of the PROFES improvement methodology is included as an appendix to the PROFES User Manual.



Chapter 2

WHAT IS THE PROFES IMPROVEMENT METHODOLOGY?

The PROFES improvement methodology is a product-driven approach to continuous process improvement. To illustrate and emphasize the importance of the *product* as a driver for improvement, it is placed in the centre of the PROFES improvement cycle in Figure 2.1. Product quality is the reason behind any improvement activity, starting with the identification of product quality needs and the determination of product quality goals. The linking element between the *product* and product development *processes* are the Product Process Dependencies (PPD), which express the cause-effect relationship between a product and a process characteristic. PPDs are used to find and determine process changes that are necessary for achieving the stated product quality improvement goals. The PROFES improvement methodology shows how to identify these dependencies and how to use them in continuous process improvement.

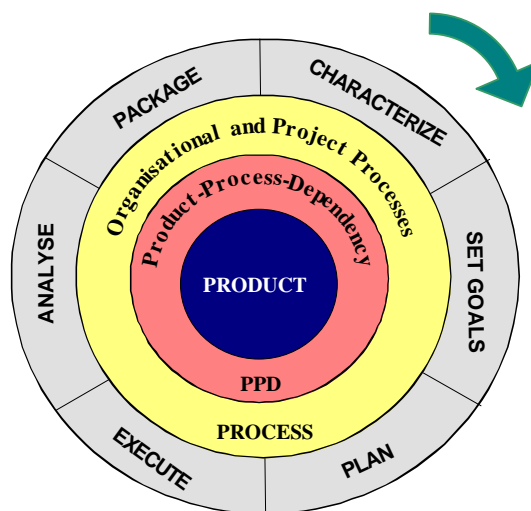


Figure 2.1. The PROFES improvement cycle

The PROCESS layer on top of the PPD layer in Figure 2.1 illustrates how product development processes may either be included in development projects, or at the organizational level. Therefore, improvement activities should include not only the project but also the organizational level. This user manual focuses on activities in development projects but occasionally deals with organizational issues, such as the setting up of a proper measurement infrastructure.

Six Phases for Managing Improvement

The PROFES improvement cycle in Figure 2.1 describes the six phases of the PROFES improvement methodology. After the six phases are completed, a new improvement cycle is begun. This section provides an overview of the PROFES phases, including the main results to be expected. A more detailed description of the phases can be found in Chapter 3, where the six PROFES phases are refined into 12 steps, with practical guidance for using the steps.

The six PROFES phases are as follows:

- **Characterize** the process improvement environment (product, processes)
- **Set Goals** for product-driven process improvement
- **Plan** process changes and implementation
- **Execute** implementation of product development project and monitoring of defined process changes
- **Analyse** measurement data and findings
- **Package** results for reuse

Characterize

When the decision to start product-driven process improvement is made, the whole organization must be prepared to carry out sustained continuous improvement. For example, top, middle, and project management must all be committed to applying the PROFES improvement methodology. This commitment should be reaffirmed every time a new PROFES improvement cycle (see Figure 2.1) is begun. Commitment by top and middle management assures the resources (money, time, human resources, etc.) necessary for carrying out improvement activities. Thus, commitment by top and middle management is vital for successful

improvement work. However, commitment by development personnel is equally important, without which any improvement activity is likely to fail.

Product-driven process improvement continues with the identification or review of product quality needs that are compiled from customer surveys, market research, or other sources. Based on these product quality needs preliminary product quality goals are set and co-ordinated with business goals. These activities form a base for forthcoming improvement activities, and can be carried out by the company itself or with the help of a consultant.

Characterizing includes assessment and descriptive modelling, for example (see Appendix 1), to determine the current capability of the processes. Prospective process changes are then pinpointed from the strengths and weaknesses of the processes. However, this is not sufficient for product-driven process improvement. The current product quality must also be understood in the terms of ISO 9126, for example (see Appendix 1), and other applicable quality factors, such as cost and time-to-market. The results of process and product characterization form the starting point for setting product improvement goals.

The main results are:

- Product quality needs co-ordinated with business goals
- Commitment of everyone involved, including top, middle, and project management
- Current capability of processes, including strengths and weaknesses
- Current product quality

Set Goals

Product quality improvement goal(s) are defined, based on product quality needs and current product quality. Using the product quality improvement goals, a related set of prospective Product/Process Dependency models (PPD models) is then identified. The PPD models contain suggestions for process improvement activities that are considered using the process capability and context information. Those suggestions that appear to have maximal effect on product quality improvement goals are then selected. If appropriate PPD models do not exist, they will have to be built. Finally, the process improvement goals are defined, based on selected process changes.

The main results are:

- Product quality improvement goal(s)
- Selected or newly developed PPD models
- Process improvement goals

Plan

Improvement has to be planned properly before project implementation. Generally, definite actions with improvement responsibilities, schedules, reporting, training, etc. are specified in improvement planning. The necessary process changes are described and modelled in sufficient detail. Goal-oriented measurement (see Appendix 1) should also be planned. The main reason for using goal-oriented measurements is to monitor and continuously analyse the effectiveness of the selected process changes on process capability and product quality. The measurement plan defines the measurement process with exact information on measurement frequency, information sources, tool support, roles involved, etc.

Please note that changes may be needed not only in the project but also at the organizational level. For example, plans for setting up the proper infrastructure need to be made if goal-oriented measurement has not been previously been carried out in the organization. This may include plans for acquisition of equipment, personnel training, definition of procedures, etc.

The main results are:

- Process improvement plan
- Prescriptive process models, including descriptions of altered processes, and related methods, techniques, tools, roles, etc.
- Measurement plan

Execute

Process improvement actions defined in the previous phases are implemented in the product development project. During project implementation, data is collected according to the measurement plan and then analysed by the project personnel who produced the data. Relevant project personnel interpret the measurement data, and produce findings that are recorded

for later use. These findings are also used to control implementation of the project, and corrective action is taken when necessary.

The main results are:

- Measurement data
- Findings (interpretations of measurement data by relevant project personnel)

Analyse

The purpose of this phase is to evaluate whether product quality has improved as planned since changes were made. The analysis phase emphasizes the gathering of lessons learnt while carrying out the planned improvement actions.

During the analysis phase, the product, process data, and findings by project personnel are thoroughly analysed and interpreted. Differences between planned improvements and actual achievements are analysed, and the root causes of any deviation are identified. The capability of processes that have either been directly changed or affected by other process changes can be evaluated, in order to help result analysis. Finally, lessons learnt are documented and relevant models (PPD models, process models, etc.) are updated.

The main results are:

- Analysed results
- Lessons learnt
- Updated and new models (PPD models, process models, etc.)

Package

Results of the analysis phase are packaged and stored for further use. The PPD models are packaged into PPD experience packages containing information on PPD model use (see Appendix 3 for examples). The process models are updated to reflect the lessons learnt. Packaging of the results in reusable form casts the basis for proper use in future projects. The availability of the packaged results throughout the organization should be ensured, for example by using the Experience Factory infrastructure (See Appendix 1).

The main results are:

- PPD experience packages
- Packaged descriptive process models
- Other packaged experience

How to start using the PROFES improvement methodology?

We recommend starting to use the PROFES improvement methodology as described in the detailed steps of Chapter 3 piloting it in selected projects. As an organization gathers experience applying PROFES it is then possible to tailor the approach to better fit the organizational context.

Experience from the industrial cases in the PROFES project show that in the beginning it is typical to emphasize assessment in the first improvement cycle for quick results. Re-application of the improvement cycle often causes a shift of emphasis to a more measurement-oriented process as historical measurement data is available.

To ensure successful application of the PROFES improvement methodology adequate mechanisms at least for quality assurance, project management and configuration management are strongly recommended.

Tailoring the PROFES Improvement Methodology

Purpose and Benefits

The PROFES improvement methodology can be tailored to determine which PROFES steps to use, and how and when to use them during a single project. Individual activities can be added, excluded, or changed if the fulfilment of the original step goals is not endangered. Selection and timing of the activities is strongly influenced by both project and organizational characteristics. It is important therefore that both analysis and subsequent implementation are carried out as soon as possible. This is important in order to be able to integrate planned improvement activities with project development plans, to gain full commitment and maximum benefit from performing them.

As the PROFES improvement methodology is an open methodology with a modular structure, it is easy to alter. The PROFES improvement methodology can be tailored to suit all kinds of projects within the organization, which helps to enhance the motivation of project members, and positively influences the results of improvement actions. It reduces the effort needed to perform these activities, and gives a better return on investment. Reuse and integration of existing practices within the organization also avoids any “re-invention” of the wheel, so to speak.

This section presents an overview on how to tailor the PROFES improvement methodology, followed by two scenarios on how PROFES could be tailored and used in practice.

Tailoring Overview

Any tailoring of the PROFES improvement methodology requires that each organization and project has its methodology analysed, i.e. documentation, methods, tools, and templates. This is then compared to existing practices within the organization, and adjusted where necessary. The tailoring procedure of the PROFES improvement methodology is presented in Table 2.1. The different phases are described on the next three pages.

Table 2.1. Tailoring overview

Tailoring procedure
<ul style="list-style-type: none">• Identify the project environment and the adaptable characteristics• Obtain and analyse input• Select PROFES phases, steps, and activities• Document the decisions and rationale for tailoring• Evaluate tailoring results

Identify the project environment and the adaptable characteristics

It is important that the project environment and adaptable characteristics are identified as soon as possible. This permits a quick start to any tailoring of the PROFES improvement methodology, and allows maximum time for the selected activities during project implementation. The most important adaptable characteristics from the organizational, methodological, and project points of view are listed below.

Organizational characteristics to consider:

- Maturity of organization
- Strength of commitment to PROFES improvement methodology
- Existing competence in PROFES improvement methodology
- Measurement programmes in use
- Ongoing improvement programmes
- Existing audit and assessment culture

PROFES improvement methodology characteristics to consider:

- Initial implementation of process modelling vs. refinement of existing process models
- Initial implementation of GQM vs. modification of existing GQM and measurements plans

- Full assessment vs. focused assessment
- Initial assessment vs. re-assessment
- Utilization of PROFES tools and templates

Project characteristics to consider:

- Lead-time constraints
- Effort (person-hours)
- Size (code volume, degree of modification)
- Defined project goals
- Defined product goals
- Significance of the project
- Functional complexity
- Ongoing project implementation phase
- Capability of the processes used
- Commitment to PROFES methodology
- Time & resources available for PROFES-dependent activities

Obtain and analyse input

In this phase, the relevant input is collected and analysed by a team made up of the parties involved, to create a feasible and practical framework for tailoring. These are similar to the identification phase characteristics, i.e. business requirements, internal organization, the project itself, and lessons learned in earlier projects. At this stage, the availability of individuals knowledgeable in PROFES improvement methodology has to be ensured either by using internal or external PROFES consultants. Sufficient training has to be planned for those project members responsible for analysing and implementing activities triggered by the PROFES improvement methodology.

Select PROFES phases, steps, and activities

The main input in this phase is the PROFES improvement methodology itself, and the methodology phases. Steps and activities will be carefully analysed and compared to existing practices in the organization. This takes into consideration the characteristics of the actual project, including impact of requirements, and lessons learned. Process improvement activities, measurement systems in use, and adapted auditing and assessment policies are examples of existing practices already applied in the organization that should also be considered.

The output will be a tailored version of the PROFES improvement methodology that defines which PROFES process steps and activities are to be used, how to use them, and when to use them during the actual development project. It is important that the analysis and subsequent implementation are carried out as soon as possible, in order to integrate planned activities with project development plans, to gain full commitment to perform them, and maximal benefits having done so.

A qualified team of PROFES methodology consultants, project and line managers, quality co-ordinators, technical experts, process owners, and developers is necessary to carry out this phase successfully. The main input, i.e. detailed descriptions of the PROFES steps and activities, can be found in Chapter 3, PROFES Steps.

Document tailoring activities and rationale

Tailoring decisions and activities should be documented together with the relevant PROFES improvement methodology relationship with reasons for adopting this approach, to enable any later analysis of the benefits of tailoring. These various activities should be carried out with sufficient effort and scheduled as a normal part of the regular project development plans. Extra time should be allocated for possible corrective action.

Evaluate results of tailoring

This last phase evaluates the success of the alterations, and documents any possible improvement suggestions. Evaluation results, together with project characteristics, performed activities, and the rationale behind the chosen modifications should be stored in the organization's experience base for future use.

Example: PROFES Improvement Methodology Tailoring in Small Organizations with Limited Experience of Software Process Improvement (SPI)

It is crucial that both managerial and project levels are totally committed to starting a software process improvement programme based on the PROFES improvement methodology (see PROFES Step 1), also in a small organization with limited SPI experience. After gaining such commitment, the following logical step is to build an organizational infrastructure for PROFES, which needs to be in place in order to carry out a PROFES improvement programme efficiently and effectively. The organizational infrastructure consists of dedicated human resources and competence, organizational support, reporting mechanisms, additional support for process assessment, and measurement programmes. Chapter 8 in the PROFES User Manual describes in more detail how to build an organizational PROFES infrastructure.

The other task to consider from the viewpoint of infrastructure is building up tool support for the PROFES improvement methodology. Appendix 5 in the user manual introduces specific tools that can be used to support different PROFES activities:

- Bootsampler – a process assessment tool
- GQMAsspect – a tool for building GQM plans
- MetriFlame – a tool for managing measurement plans and data
- Trend Analysis – a tool for monitoring process capability

By this stage, the results are usually as follows. A PROFES team has been established, PROFES resources have been allocated, management reporting processes have been defined, training and promotion organized, support for process assessment, measurement programmes, and tools increased. The following step is then to tailor the PROFES improvement methodology by carefully analysing the PROFES steps, activities, and available templates, in order to decide how and when to use them, while taking the organizational and project characteristics into consideration.

A typical result of analysis of a small organization is described below in terms of decisions and rationale regarding tailoring.

Example: Tailoring Decisions in a Small Organization with Limited Experience on SPI:

- A general decision was made to use a simplified PROFES improvement methodology for the actual development project, concentrating only on core product quality characteristics from the customer's point of view, and core processes with the most influence on attaining defined goals.
- Due to lack of any greater knowledge of PROFES and previous experience in SPI, a PROFES consultant begins the processes, new from the organizational viewpoint, and provides support during their implementation.
- Support from a PROFES consultant is crucial during certain activities in PROFES Steps 1 to 5. These include motivating and training management and project members, identifying core product quality characteristics and needs using the ISO9126 standard, assessing the capability of the processes in use, and in defining the product improvement goals.
- The PPD concept and repository are used to determine the process changes necessary. The changes suggested are ranked and only those with the greatest estimated effect on final product quality are documented, and carried out during the actual project.
- Supported by the PROFES consultant, the GQM approach is pilot-tested performing the activities described in PROFES Step 8. This approach results in a limited GQM and measurement plan, concentrating only on a small set of measurement goals, and measurements reflecting the processes chosen from PROFES Step 6, and defined product improvement goals.
- All improvement actions are scheduled and resources allocated in the project development plans, and their progress monitored in normal project follow-up meetings, PROFES Step 9.
- Implementation takes place according to the integrated development plan. Measurement data is collected according to the GQM plan. The data is processed in advance, and analysed together with project members at GQM feedback sessions. Corrective actions are taken if necessary based on the analysis results, PROFES Step 10.
- Measurement results are evaluated and identified, and tested PPDs are supported, modified, or rejected. Lessons learned are collected for use in future projects, PROFES Step 11.
- A simple manual experience base, such as an "experience binder", is created for collecting lessons learned, PPD experience, GQM plans,

and achieved results, together with the project characteristics. The PPD repository is updated with any necessary modifications to the existing PPDs or by adding new ones, PROFES Step 12.

Tailoring rationale:

In a small organization with limited SPI experience, using the PROFES improvement methodology as a base for all activities was essential, as the organization had neither sufficient experience nor established its own practices in the areas defined. The decision to hire a PROFES consultant to provide the missing competence and the greater effort required at certain stages was crucial for the success of the improvement programme. This is a simplified approach that concentrates solely on core product quality characteristics and processes from the customer's viewpoint, and is expected to have the most influence on achieving goals set, motivating project members and line managers, and improving return on investment.

Example: Adapting the PROFES Methodology in a Mature Organization

This example describes adapting the PROFES improvement methodology to suit a mature organization, which has already applied process improvement activities for continuous product quality improvement. A measurement system and an internal assessment framework built on the CMM (Capability Maturity Model), for example, would already be in place in this scenario. The main purpose of tailoring in this case is to reuse existing improvement practices as much as possible, and to refine them when applicable. Analysis results and mapping of the PROFES methodology, steps, and activities are described below in terms of tailoring decisions and their rationale.

Example: Tailoring Decisions in a Mature Organization using CMM:

- Instead of performing PROFES Steps 2,3, and 5 for product quality, the product goals defined in the annual process improvement programmes are used instead.
- The current process capability of PROFES Step 4 is determined using a CMM assessment framework.
- The PPD concept in PROFES Step 6 will only be used as a support method for refining and prioritizing planned process improvement activities.
- Instead of using full GQM scope in PROFES Step 8, existing measurement programmes will be updated with selected measurements from refined and prioritized processes.

- Improvement planning will be integrated into normal project planning with respect to time and resource allocation, and documented in the project management reports, PROFES Step 9.
- GQM-type feedback sessions will be used to effectively communicate and analyse measurement results, to agree on corrective actions, and to collect lessons learned, PROFES Steps 10 and 11.
- Achieved results and lessons learned will be stored in a experience database, together with the project characteristics, thus enabling automatic data retrieval, PROFES Step 12.

Tailoring rationale:

It was necessary to take a simplified approach to adapting the PROFES improvement methodology, due to the mature practices already existing in the organization. The integration of certain PROFES steps into established improvement practices would reduce the time necessary for carrying out improvement actions. This can be seen as a desirable result, given the constant need to reduce lead-time.

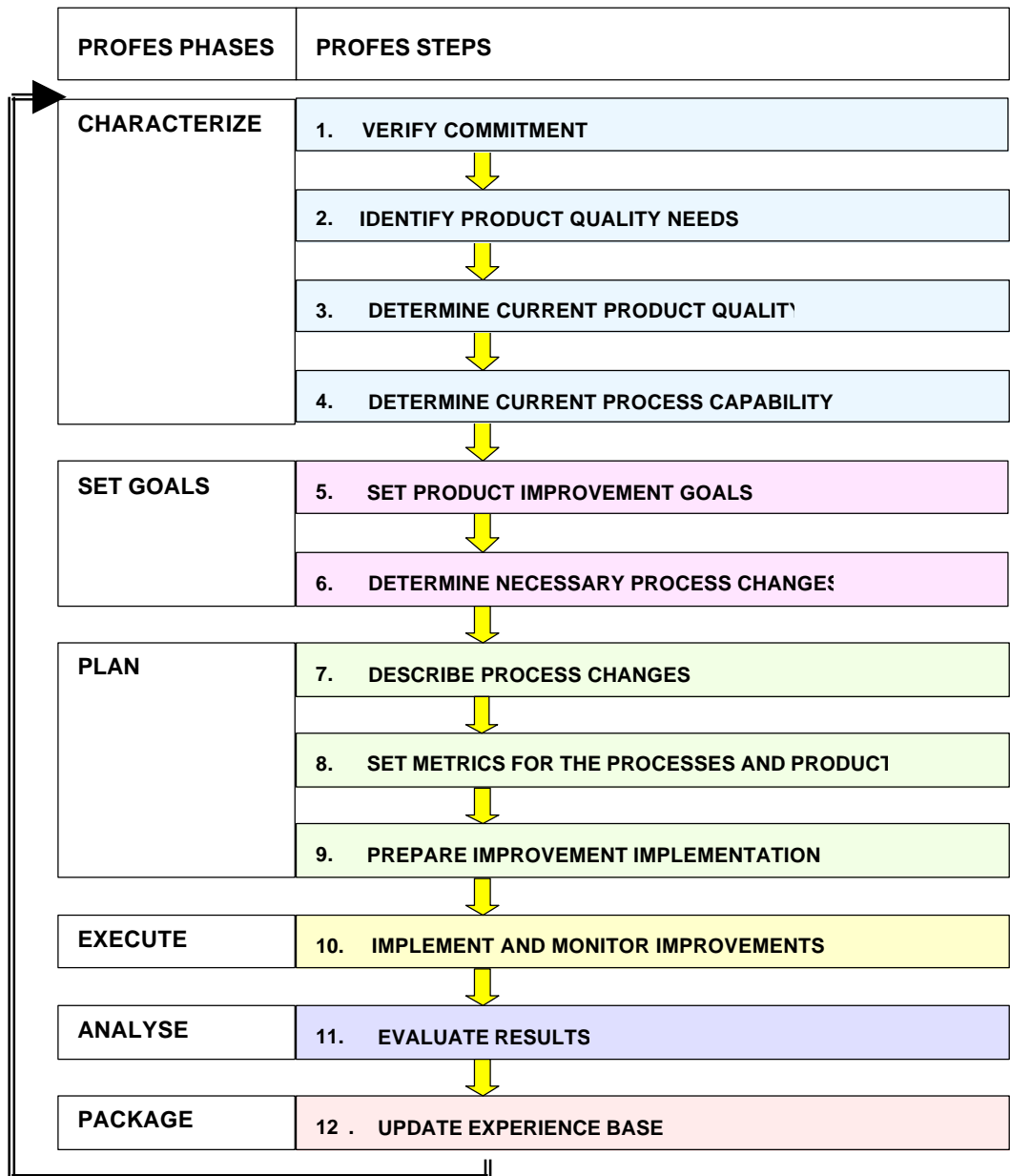


Figure 3.1. The phases and the steps of the PROFES improvement methodology

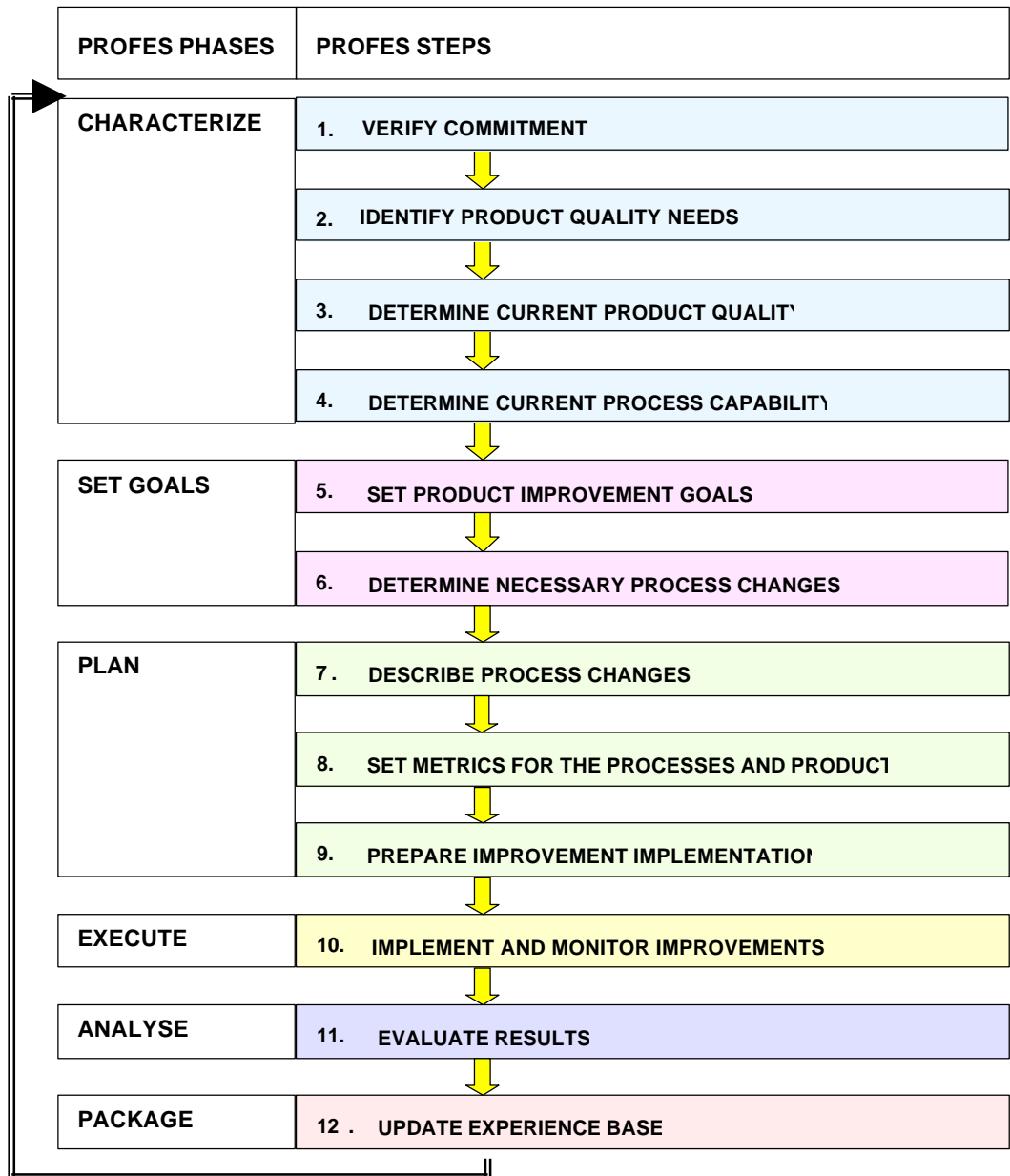


Chapter 3

THE PROFES STEPS

In this chapter, we will describe the 12 steps of the PROFES improvement methodology in detail. Our purpose is to assist you in using PROFES in practice. At the beginning of each step, the purpose of the step and the goals to be reached during the step are described. All activities are described with examples and detailed instructions. An estimate of the average duration of the step and the effort it requires are also given. Tools and templates that can be used to support PROFES steps and activities are described and further reading is referred to. The work products that each step uses and produces are described, and resource allocations with roles, responsibilities, and expected effort are given. The figures for expected effort are presented to help to understand the magnitude of the work involved, but actual values will vary depending on the context. The roles for carrying out PROFES improvement are divided into managerial, expert, and supporting roles that are further divided into sub-roles. Please note that one person can assume many roles. The number of people needed for the PROFES team depends on the size of the company and its product development. For detailed information about establishing a PROFES team, please refer to Chapter 8 on building an infrastructure for sustained process improvement in an organization.

The PROFES phases and steps are presented in Figure 3.1 in the form of a flowchart. In the following sections, the main procedure steps are described in more detail.



STEP 1	Verify Commitment	PROFES
	<p>1-1 Identify organization's business needs and improvement objectives</p> <p>1-2 Motivate top and middle management</p> <p>1-3 Motivate project members</p> <p>1-4 Define organizational context</p> <p>1-5 Define overall plan and schedule</p>	

Purpose

Commitment is very important for successful improvement results. Improvement actions can be carried out efficiently only if the appropriate business goals, personnel issues, and specific company characteristics are acknowledged and addressed at all levels of the organization. Only with managerial commitment can the resources (time, money, and people) necessary for improvement work be secured. All people whose work will be affected or changed due to improvement activities should also be committed to improvement.

When a new improvement cycle begins the commitment of top and middle management and project members has to be verified. The improvement objectives that trigger an improvement cycle have to be understandable, challenging, and relevant. Strategies for achieving improvement objectives should be understood and accepted by everyone. Objectives should be reviewed regularly and must reflect any change in the organization's needs. The organizational context sets the framework for improvement action and the benchmarking and reuse of its results. Commitment should be re-established every time situational changes occur that concern people, activities, and company objectives.

Step goals

- The organization's business needs and improvement objectives for product and process quality are identified.
- Product quality characteristics, ongoing improvement initiatives, and their priorities are identified.
- Commitment of top and middle management is verified.
- Commitment of project members is verified.

- Contextual information of the organization and projects is defined.
- An overall plan and schedule for improvement activities is defined.

Activities

Firstly, it should be clear why an improvement cycle is started. Secondly, the top and middle management need to be committed but also project members need to be convinced of the new improvement initiative. The organizational context needs to be defined to aid further focusing and reuse activities. Finally, an overall plan and schedule is drafted and agreed to.

This step therefore consists of the following activities:

- Identify the organization's business needs and improvement objectives
- Motivate top and middle management
- Motivate project members
- Define organizational context
- Define overall plan and schedule

Activity 1-1	<i>Identify the organization's business needs and improvement objectives</i>
	<ul style="list-style-type: none">• Identify organization-specific needs and objectives• Identify product and process quality improvement needs• Identify improvement initiatives <p>This activity begins by focusing on the organization-specific needs and objectives, which drive both product and process as well as their improvement. Organizational needs and objectives drive the identification of product and process quality improvement needs. For example, product x must become market leader or the lead-time of product y must be reduced by 30%. These can be determined through interviews and discussions with company management and project managers. The discussions should cover customer viewpoints as well as market</p>

scenarios, business expectations, future product development trends, and improvement initiatives. The improvement initiatives and programme discussions should also cover all previous, current, and future initiatives of the organization, and should be documented.

**Activity
1-2*****Motivate top and middle management***

- **Brief top and middle management**

The commitment of top and middle management personnel is secured through motivation presentations and management briefing. The presentations briefly review the advantages of the PROFES improvement methodology and its main features, as well as results and benefits expected from improving the organization's product and process qualities. The main point is to emphasize the specific gains the organization may achieve by using the PROFES improvement methodology. The style of the presentations should be appropriate for the level of executive management. The presence and participation of top management at PROFES briefings is desirable, as it also makes it much easier to secure the commitment of middle management and technical personnel.

**Activity
1-3*****Motivate project members***

- **Brief project personnel**

The commitment of project management and members should be secured through motivation presentations and PROFES briefing. The focus of these presentations is more project-oriented – how the improvement initiatives will help the project to achieve its goals, and how the PROFES improvement methodology is applied in practice.

**Activity
1-4*****Define organizational context***

- **Characterize the organizational and project context**

Contextual information is collected and the target organization, its projects, the methodology used, and the tools and techniques are classified. This information can then be used later for process assessment, product-process dependency model building, reuse, experience packaging, and benchmarking.

**Activity
1-5*****Define overall plan and schedule***

- **Define overall plan**
- **Define overall schedule**
- **Get acceptance for the overall plan and schedule**

An overall plan and schedule is drafted based on the available information gathered during Step 1. Management needs to agree to the overall plan and schedule. Thus, Activity 1-5 contributes to the overall goal of the step – Verify Commitment. The plans will be updated in more detail as the specific operational priorities for the improvement activities will become clearer in later steps.

Average duration and effort

The total duration of this step is around one month.

Tools and templates

- PROFES presentations for motivating management (Management briefing)
- PROFES presentations for motivating project members (PROFES briefing)
- Template for the process improvement plan (See Appendix 2)

Work Products

Input work products	Output work products
<p>Organizational level:</p> <ul style="list-style-type: none"> • General organizational information • Business goals • Customer survey results • Market research results • Customer feedback • Organizational context information <p>Project level:</p> <ul style="list-style-type: none"> • Project environment specifics • Project context information • Product development goals 	<p>Organizational level:</p> <ul style="list-style-type: none"> • Commitment of top and middle management • Preliminary product and process improvement needs • Organization's classification • Overall improvement plan <p>Project level:</p> <ul style="list-style-type: none"> • Commitment of project management and members • Project classification • Overall improvement plan

Resource Allocation

Roles, responsibilities and requested skills

Managerial roles

Top and middle management

The main role of management is to justify and provide resources for starting improvement activities.

Expert roles

PROFES expert(s)

PROFES experts are responsible for presenting and marketing PROFES concepts and methodology to the organization, and help to prepare overall plan for the improvement cycle.

Project members

Project members, including project management and project technical personnel, are involved presentations and technical discussions, if needed.

Support roles

No specific support roles defined.

Expected effort and role*PROFES experts*

The actual effort spent on this step is focused on motivating top and middle management. The time taken by presentations is about 4 hours per presentation. The time required for preparing overall plan depends on the extent of the improvement initiative.

Top management

In this step, effort is spent on briefings and negotiations with the PROFES consultants, whether internal or external.

Middle management

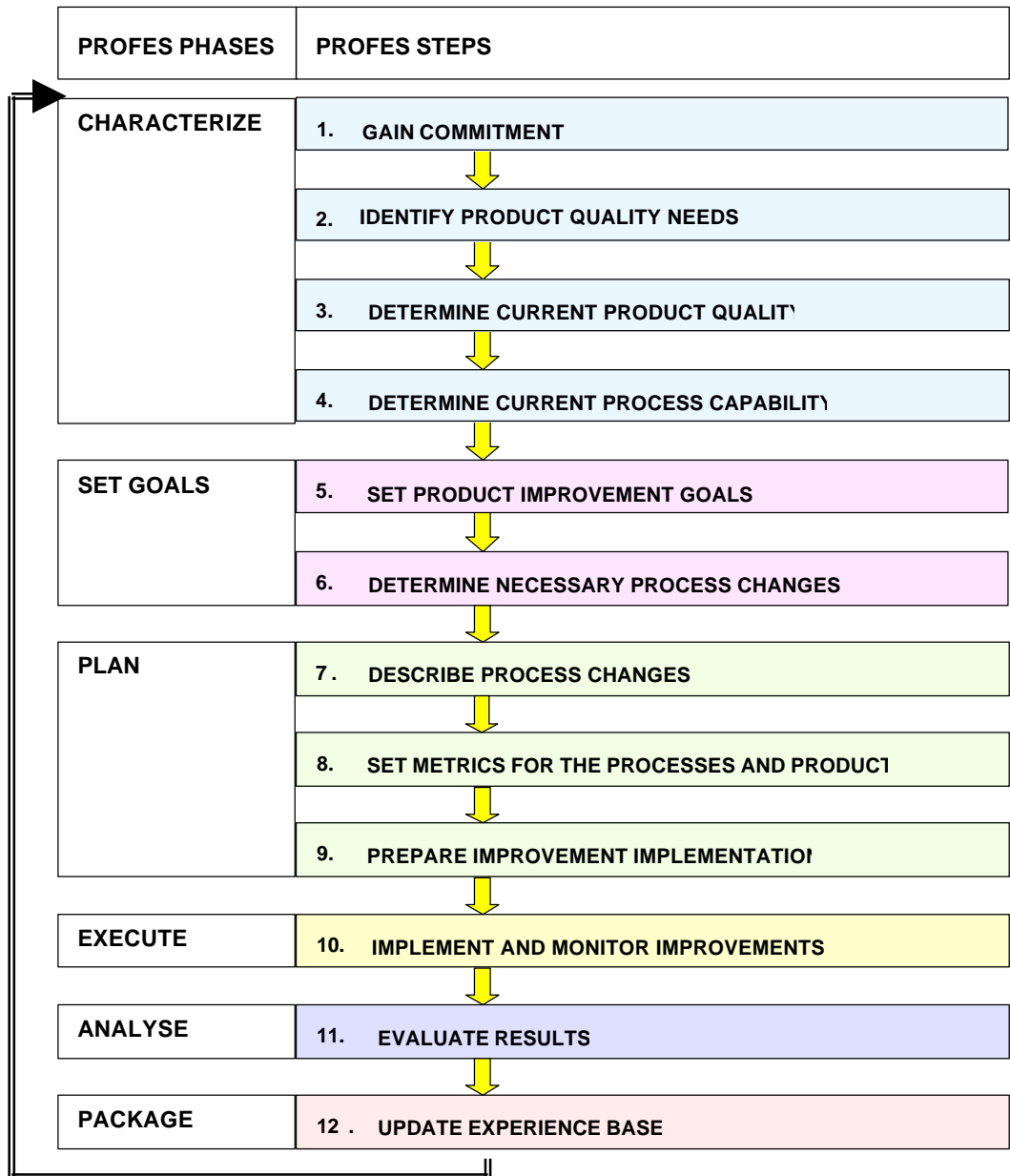
Middle management spends less than 4 hours on this step.

Project members

Project members spend less than 4 hours on this step.

Methods and techniques

No specific methods and techniques are prescribed.



STEP**2****Identify product quality needs****PROFES**

- 2-1 Survey product quality needs**
- 2-2 Document product quality needs**
- 2-3 Set preliminary product quality goals**

Purpose

Product quality needs must be identified and understood. Based on these needs, product quality targets for development can then be set. These targets form a basis for product quality goal selection and prioritization. Product quality needs are transformed into preliminary product quality goals.

These preliminary product quality goals are also used to focus the process assessments of Step 4 (Determine current process capability), and are also used to set product improvement goals in Step 5 (Set improvement goals).

Step goals

- Product quality needs are known and presented in the form of a product quality profile
- Preliminary product quality goals are set

Activities

In order to identify product quality needs, product stakeholders are interviewed, and their needs for product quality are categorized. In order to do this, those stakeholders involved with each product should be identified. This step therefore consists of:

- Performing a survey of product quality needs
- Documenting the results of the survey
- Defining the preliminary product quality goals for use by subsequent steps

**Activity
2-1*****Survey product quality needs***

- **Identify product stakeholders**
- **Schedule and invite representatives of stakeholders to an interview**
- **Hold interviews**
- **Document interviews**

Identify and invite product stakeholders

After collecting preliminary product improvement needs in Step 1, more information can be obtained by interviewing product users. The term user not only refers to the actual users of the product, but also to anyone who is involved in specifying requirements, such as people from marketing, development, manufacturing, etc. We call them stakeholders. This survey must be organized in such a way that all stakeholders interested in the product are either consulted or represented. Some form of business or domain modeling is applied to determine all the stakeholders for a product.

Internal stakeholders should also be included, i.e. stakeholders within the organization that have specific demands for a product. For example, the manufacturing department that installs software in the product has demands regarding the release of the software and the corresponding installation documentation. Maintenance and operation support people should also be considered, as they have specific demands regarding product quality, and are often also very familiar with user preferences and requirements.

Each selected stakeholder is sent an invitation to an interview. This invitation may indicate the topics for discussion beforehand.

Hold interviews

In the open interviews for the survey, there are basically five topics for discussion:

- List the exact tasks performed

This first topic is discussed to gain more insight into how users work with the product. Interviewees are asked to list the exact tasks performed when handling the product. This provides an overview of the product's context of use.

- Effort needed for each task

The effort spent on these tasks is determined to provide an overview of the amount of time required.

- Previous quality problems

Interviewees are asked to recall any quality problems with the product, previous versions of the product, or other similar product types.

- Expected quality problems

Based on their experience and current understanding of the product, interviewees are asked whether they expect specific quality problems.

- Existing high quality products

The final topic deals with the product's specifically good characteristics. Interviewees are asked to give examples of good qualities that the product should possess. This is done simply to counterbalance the preceding questions, which discuss bad quality in need of improvement. Good quality must also be dealt with, in order to make sure that its level of quality is maintained.

Document interviews

Users express their wishes and requirements in their own terms, and rate them in order of importance. We want them to express their wishes in their own terms so that it must be quite clear what those needs are for specific contexts of use. Translating these requirements into more generic terms is another matter, and will be addressed later. A second point of discussion is for providing quantifiable criteria to create metrics for each wish that determine how well a product meets its requirements. It is not sufficient to simply state a demand, as whether or not it has been met is difficult to check. Users sometimes also speak in 'solutions,' and ask for specific process solutions. However, it is better for users to speak in 'problems' or 'product goals' and leave the solving of these problems to the designers. Their solutions have to be sufficient, and therefore we need to have measurable criteria that can be checked and managed. Metrics can be collected to evaluate a specific requirement, which must be guided by a wanted value. This indicates the value that the metric must have in order to fulfil the requirement. Metrics and wanted values should be retrieved from the interview and confirmed by the interviewee when reviewing the interview report.

**Activity
2-2*****Document product quality needs***

- **Translate stakeholder wishes into generic terms**
- **Make a product quality profile**

Interviews conducted with all the selected representatives of typical and/or the most important users result in a list of wishes and requirements in the users' own language. They must then be translated into generic terms that developers can work with. PROFES suggests using the ISO 9126 terms enhanced with cost and time-to-market. ISO 9126 classifies product quality by six product quality characteristics: functionality, reliability, usability, maintainability, efficiency, and portability. Furthermore, each characteristic is further refined into several sub-characteristics.

Firstly, for each user wish/requirement the quality characteristic referred to should be identified and secondly, the respective quality sub-characteristics. This translation into ISO9126 terminology is an important step towards a generic product quality model, and is also not very difficult once the ISO9126 definitions have become fully familiar.

The ISO9126 definitions are:

Functionality – the capability of the software to provide functions that meet stated and implied needs when the software is used under specified conditions.

Reliability – the capability of the software to maintain the level of system performance when used under specified conditions

Usability – the capability of the software to be understood, learned, used, and liked by the user when used under specified conditions.

Efficiency – the capability of the software to provide the required performance relative to the amount of resources used under stated conditions.

Maintainability – the capability of the software to be modified.

Portability – the capability of software to be transferred from one environment to another.

Next, each stakeholder requirement has to be evaluated by the project manager, as everything can not be realized, as other objectives such as time and cost-effectiveness also play a part. A meeting should be held during which all product are discussed with the project manager, and all requirements are either accepted, rejected, or held pending.

These requirements and their target values need to be documented. We recommend visualizing these product quality targets in a product quality profile, which displays the targets according to ISO9126 dimensions. A product quality profile example is shown in Figure 3.2. In this example figure, 'wanted value' indicates the product quality requirements that were stated by all stakeholders. 'Target value' indicates the target that the project manager has set for the project. The scale in the figure is a modified ordinal scale of increasing requirements for reliability ranging from 'D' to 'A'. The project manager also recorded 'wanted value' and 'target value' for cost and time-to-market.

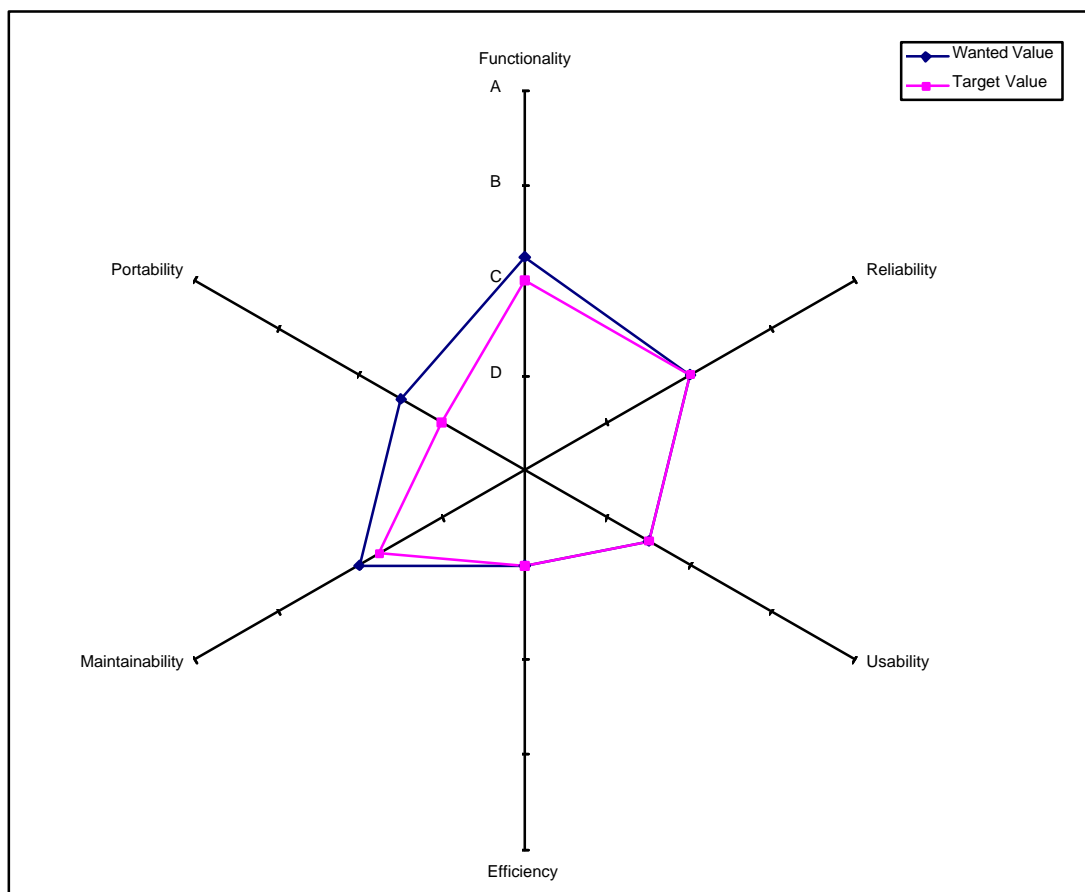


Figure 3.2. Example product quality profile

**Activity
2-3*****Set preliminary product quality goals***

- **Identify the product quality areas that require improvement**
- **Prioritize the product improvement areas**
- **Select the preliminary product quality goals**

Product quality needs are transferred to preliminary product quality goals that reflect possible areas for further improvement. Based on the product quality targets classified according to ISO9126 (sub)-characteristics, those areas requiring further attention should be prioritized, and made the responsibility of the project manager. With this list of priorities assigned to the product quality characteristics, a set of preliminary product quality goals is made available.

Average Duration and Effort

The amount of time necessary to complete the whole step is approximately 2 weeks.

The amount of time necessary strongly depends on the product itself and the schedules of the interviewees. Scheduling the interviews is a critical point, and we recommend that interview appointments are made as soon as possible, as the (often busy) schedule of interviewees can significantly delay Step 2.

Tools and Templates

- ISO 9126 can be used as checklist,
- Structured interview techniques, see for example: Rini van Solingen and Egon Berghout, *"The goal/question/metric method, a practical method for quality improvement of software development"*, McGraw-Hill ISBN 007-709553-7, 1999.

Work Products

Input work products	Output work products
<ul style="list-style-type: none"> • Customer survey results • Market research results • Customer feedback • Business goals • ISO9126 • Preliminary product quality needs 	<ul style="list-style-type: none"> • Product quality needs • Product quality profile • Preliminary product quality goals

Resource Allocation

Roles and responsibilities

Managerial roles

Project Manager

The project manager is involved in reviewing all the needs specified for the product. The project manager has the responsibility to decide whether each product need will be accepted, rejected or held pending. Furthermore, the project manager is responsible for setting the preliminary product quality goals, and prioritization of the quality areas.

Expert roles

PROFES expert

The PROFES expert takes care of all activities necessary to create the deliverables, including interview preparation and reporting, also including domain modelling and documenting the product quality profile. Special skills are required regarding the interviews. Holding an interview and handling stated quality needs is often a difficult task.

Support roles

Interviewees

The interviewees, or users, are persons involved in specifying requirements. Their main activity is to provide information during the interviews, and review the interview report for mistakes, inconsistencies or missing requirements.

Expected effort/role

PROFES expert

Conducting the interviews for the extensive survey takes approximately 8 hours in all, distributed between preparation, the interview itself, and processing the results.

Stating the priorities for product quality characteristics in order to define the preliminary quality goals takes less than two hours.

Interviewees

The interviews for the survey take a maximum of two hours, depending on the knowledge of the interviewee. Preparation for interviews is not obligatory, but an interviewee can consider basic quality requirements beforehand, based on the interview agenda. Another two hours is needed to review the interview report on incompleteness and inconsistencies.

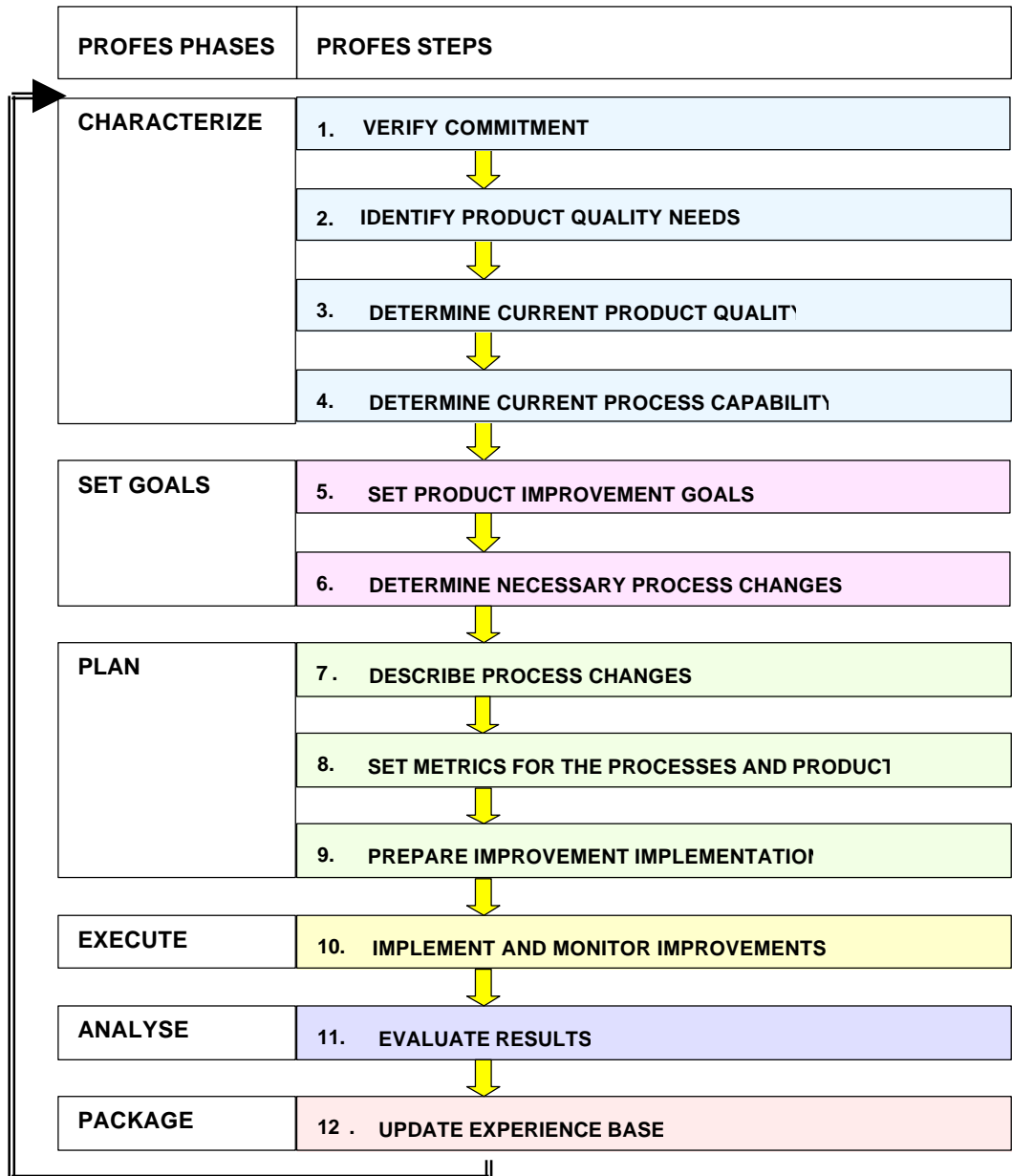
Project Manager

Besides his activity as an interviewee, the project manager will also have to review all product requirements, and set the preliminary product quality goals, 4 hours work in all.

Methods and Techniques

Multi-Party Chain techniques: 'User-perceptions Of Embedded Software Quality', R. Kusters, R. v. Solingen, J. Trienekens, Chapter 4 (pp. 148-163) in: Software Quality from a Business Perspective - Directions and advanced approaches, Kluwer Bedrijfs Informatie, ISBN 90 267 2631 7, 1997.

More information on MPC can be downloaded from the internet-page: <http://www.tm.tue.nl/vakgr/it/mwerkers/rso/rini.htm>



STEP**3****Determine current product quality****PROFES****3-1 Acquire product quality data****3-2 Evaluate current status of product quality****Purpose**

To determine the current status of product quality, for comparison with the output result of Step 2 (product quality needs). The differences between current and needed product quality are used in making reasoned plans for carrying out product and process improvement actions. This step is also important to determine those product qualities that are much better than necessary.

Step goals

- Current status of product quality is known

Activities

Identification of current product status in order to determine the current product quality is carried out with product evaluation and more demanding product measurement and/or assessment techniques, which are selected as required.

The following activities are included in Step 3:

- Acquire product quality data
- Evaluate current status of product quality

If no version of the product is available to the project, this step will be quite difficult to carry out. In some cases it might be reasonable to focus on the previous generation of the product, but this does not always make sense. Estimating the product quality that is likely to be delivered by the current process is also an option, although this is not supported by the PROFES improvement methodology.

**Activity
3-1*****Acquire product quality data***

- **Prepare product data acquisition**
- **Acquire product quality data**
- **Report product quality data**

The purpose is to collect data from the product regarding the product characteristics (e.g. ISO 9126), with the help of product measurement and/or a product characterization questionnaire. This data will be used to identify which product areas are already satisfactory, and which ones require additional work. The questionnaire examines the product's important quality characteristics. PROFES does not provide generic questionnaires for this activity. We recommend that one is made for every product type being developed. The results of Step 2 can be used to set up such a questionnaire.

Existing measurement data can be also used here. Parts of the necessary product information may be achieved using data from earlier measurement programmes, measurement of similar products or older version of the product. Product information can also be derived from verification and validation activities performed during past development cycles.

In Step 2, an survey of product quality needs has been made in which product quality needs were specified, translated into ISO9126 terms, and enhanced with measurable criteria. These criteria were specified in the form of metrics and the required value was specified as well. In this step, we suggest that data be collected for these metrics and a check made whether the current value for these metrics is in line with the required value specified in Step 2.

Measuring the data for these product quality metrics can be done by the quality engineer who manages the project. However, it may be necessary to involve a member of the project team as well. Specific metrics may possibly not be calculated yet, due to the current development status of the product. Estimates by the project team are sufficient, as long as they do not know the required value before estimating it. Project team members have a tendency to align their estimates with targets, which should be taken into account.

Before carrying out a product evaluation, proper preparation is necessary. This involves inviting people, assembling equipment, installing testing environment, setting up product evaluation procedures, etc. Preparation of

product quality data acquisition is the most important and time-consuming part of the work. Once all the pre-requisites are arranged, data collection becomes quite easy. The findings should be reported in a product quality data report, which reports the current values to the metrics and requirements set in Step 2. An example is provided in Table 3.1.

Table 3.1. Example product quality data

Number	Wish	Quality	Question/Metric	Target	Current
84	The product has to be usable	Usability	Do users find the UI clear?	Yes	Yes
87	No usability problems should occur	Usability	# bugs with the status: Open and severity: Usability (as recorded in Defect tracking tool)	Max. 15	35

Activity 3-2	<i>Evaluate current status of product quality</i>
-------------------------	--

- **Analyse product quality data**
- **Discuss product evaluation results**
- **Determine current product quality**

Based on the values of the questionnaire or the values that were measured during Activity 3-1, the current status of product quality can be determined. The result should be documented according to the ISO9126 quality (sub)-characteristics. This overview of current product quality status must indicate:

- What is the wanted product quality?
- What is the actual product quality?
- What is the difference there between wanted and actual quality? Is the quality excessive?

These differences are of especial interest, because they indicate which product quality areas the process is not sufficiently effective. This information can be very helpful in focusing the process assessment in Step 4. It is obvious that the areas in which improvement is needed are where the actual product quality is not acceptable. Therefore the output of Step 3 is also be highly relevant for Step 5, in which the product improvement goals will be set. It is also important to identify which product areas have a higher quality level than necessary.

The results of such an evaluation can be demonstrated by a diagram similar to that in Figure 3.3.

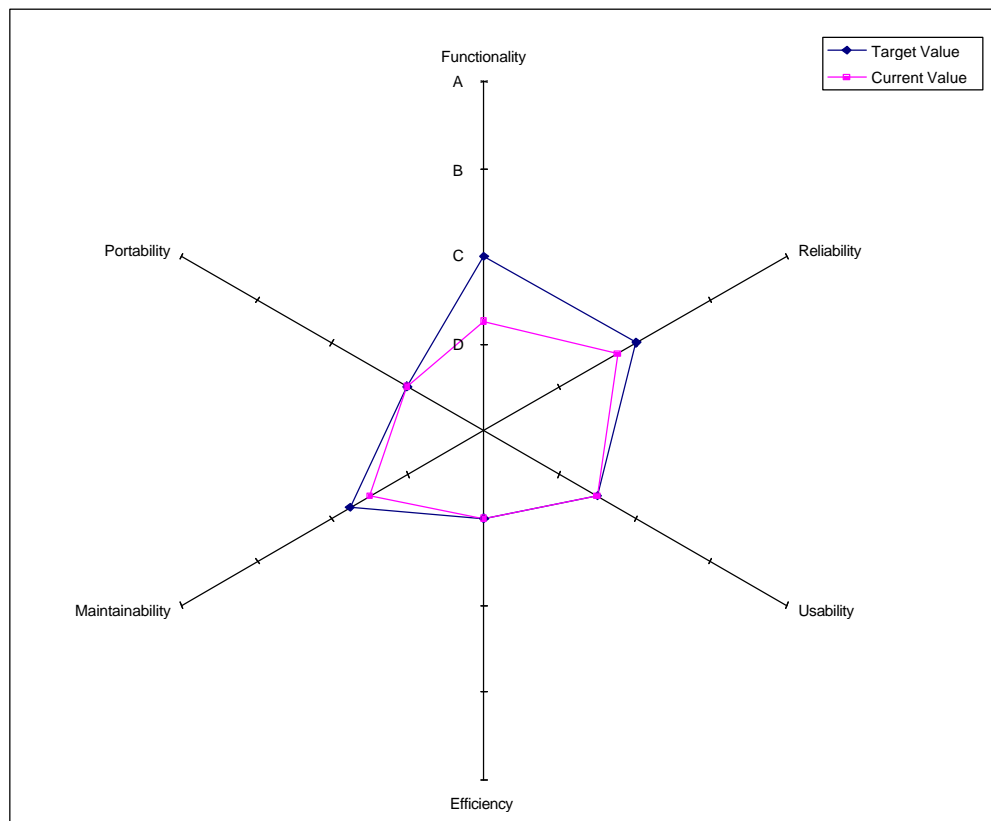


Figure 3.3. Example result of evaluating product quality status

Based on such figures, it becomes possible to determine in which product quality areas improvements are required. In cases where the current product quality is lower than the target, improvements are particularly required.

Average Duration and Effort

The total duration and effort for the identification of current product quality will be about two person-weeks, depending on the specific product and the degree of difficulty in collecting the required data.

Tools and Templates

Product evaluation and measurement tools can provide suitable support for the step. MetriFlame can provide support during the analysis and representation of the measurement results. Other software tools can be used to calculate specific metrics on the software code, for example the number of source lines, or cyclomatic complexity.

Work Products

Input work products	Output work products
<ul style="list-style-type: none"> • Application domain characteristics • Measurement data • ISO9126 • Product quality profile • Experience base 	<ul style="list-style-type: none"> • Current status of product quality

Resource Allocation

Roles and responsibilities

Managerial roles

Management

Management should be consulted on decisions for product quality. If not possible, management will at least receive the outcome of these decisions.

Marketing manager

Decisions on product quality levels are mainly taken in close co-operation with the marketing department. A marketing manager should be consulted during the decision process of what the product should or should not do.

Project manager

The project manager should be consulted during the activities to establish the current product quality. This effort will remain limited.

Expert roles

PROFES expert

The PROFES expert carries out most of the activities in Step 3, and is responsible for product evaluation and determination of

the current status. The PROFES expert produces the deliverables from this step.

Support roles

Software Engineer

It might be necessary to consult one or more software engineers during product evaluation of the current product. The software engineer will support the quality engineer where necessary. However, the role of the software engineer will be limited.

Expected effort/role

PROFES expert

The effort spent by the PROFES expert on the above activities will be about 24 hours for the product data acquisition, depending on the difficulty of setting up the testing environment, and 16 hours on the evaluation of current product quality and corresponding reporting tasks.

Project manager

The project manager will spend less than four hours on this step.

Software Engineer

The software engineer will spend less than eight hours on this step.

Marketing manager

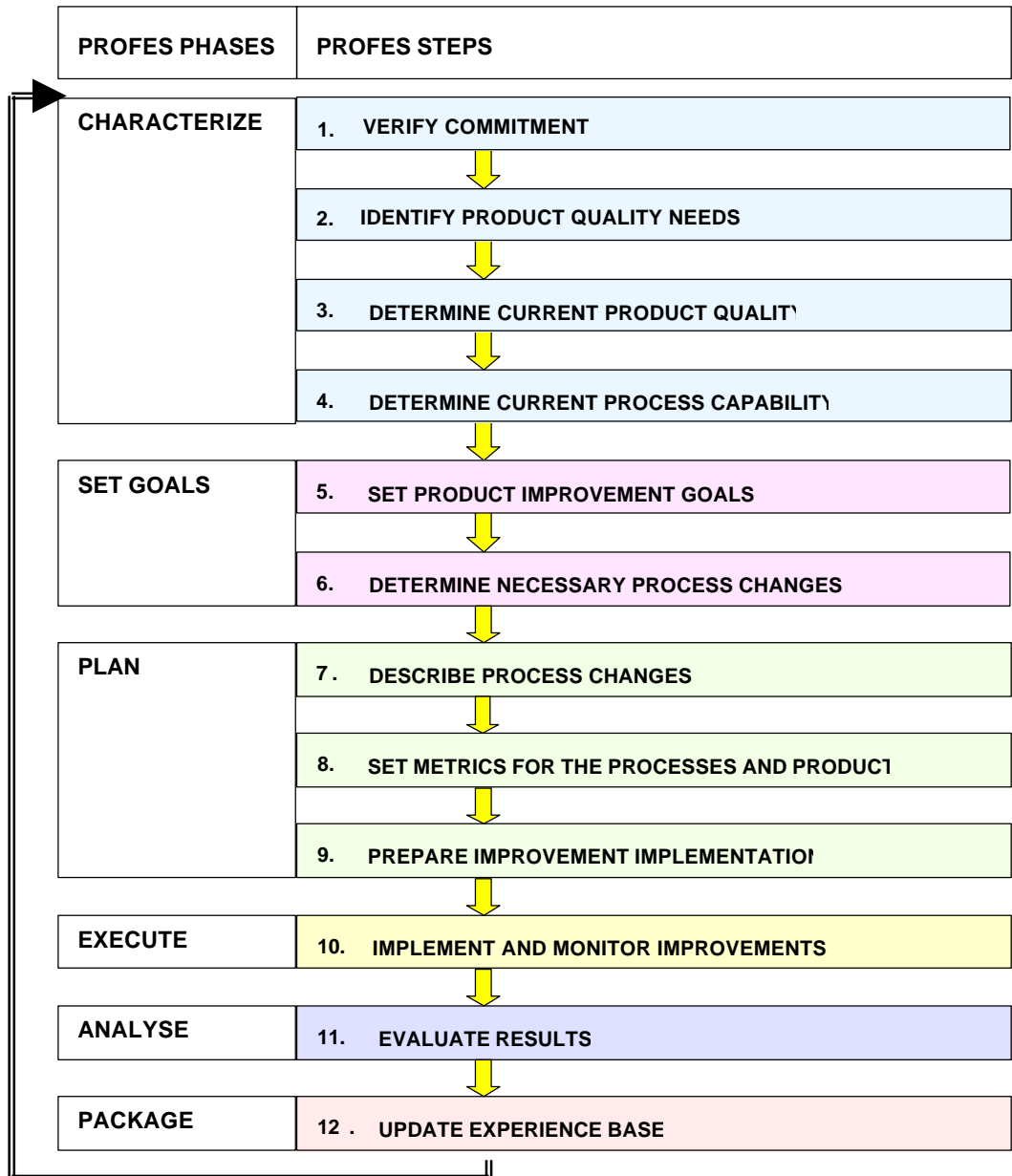
The marketing manager will spend less than four hours on this step.

Management

Management will spend less than four hours on this step.

Methods and Techniques

No particular methods and techniques are provided for this step. Product quality metrics are measured, analysed, and comparing to the requirements set in Step 2.



STEP 4	Determine current process capability	PROFES
	4-1 Preparation	
	4-2 Execution	
	4-3 Reporting	

Purpose

All process improvement activities should be based on a clear understanding of the context and status of current software processes. The purpose of Step 4 is to evaluate the capability of the current software processes by analysing existing process and project documentation and interviewing key personnel. Documented software processes are analysed and undocumented processes outlined by using appropriate process modelling methods and techniques. Process descriptions are used to clarify what software activities have been recognized in the organization. The major activity of the step is to assess the process capabilities. Existing process descriptions, previous assessment results, and measurement data are used as a starting point of the step.

Please note:

- The chosen assessment method should support both organizational (SPU) and project level assessments.
- See Chapter 4 for integrating assessment and measurement planning activities.

Step goals

- Current process capability is determined
- Process improvement recommendations are documented and communicated

Activities

Process assessment is used to evaluate software process capabilities, in order to identify candidates for process improvements. The process is assessed at both organizational level (SPU-Software Producing Unit) and project level. At the organizational level, the goal is to assess the processes as defined and implicitly agreed by the organization, and in project level assessment, the goal is to assess how these processes are performed in practice.

Step 4 includes three activities:

- Preparation
- Execution
- Reporting

Activity 4-1	<i>Preparation</i>

- **Collect and analyse process and performance documentation**
- **Recognize and describe current software process**
- **Plan and schedule assessment**

Preparation for software process assessment is begun with the collection of documentation on the company procedures, software processes, quality system, projects, and measurement results. This material is carefully analysed, and the analysis forms the basis for outlining the current processes and planning of the assessment execution.

The collection and analysis of process and performance documentation is used to acquaint the PROFES team with the organization and the application domain in advance. It is based on a study of all the relevant documentation that is collected in this step. The documents include for example:

- Quality manuals
- Process descriptions
- Possible previous assessment reports
- Possible results of previous measurement programmes
- Project documentation
- Product documentation

In **Recognize and describe current software process**, explicit process descriptions are analysed and implicit processes are identified and outlined using appropriate process modelling techniques. In these descriptions, the following items should be recognized:

- Artifacts consumed and produced
- Activities carried out
- Agents (with roles) involved
- Tools used
- Technologies, techniques, and methods used
- Relationships between activities and artifacts, i.e. flow of artifacts
- Assignment of roles to activities
- Application of technologies/techniques/methods in activities
- Relationships between products, i.e. product hierarchies
- Relationships between roles, i.e. communication network.

In **Plan and schedule assessment**, the PROFES team carries out negotiations with the organizations about their needs and priorities. An activity schedule is defined together with the sponsoring organization. It includes a plan containing an overall time schedule of the assessment activities.

Identification of the interviewees should be included in the plan, and interview invitations including the topics of the interview sent to the relevant persons.

Activity 4-2	<i>Execution</i>
-------------------------	-------------------------

- **Hold opening briefing**
- **Collect findings through interviews**
- **Assess process capabilities**
- **Provide feedback**

In this activity, information on the organizations' processes are collected by performing interviews of key personnel and evaluating available documents, as planned in the Preparation activity. The information is collected at organizational level as well as at project level. Qualified assessors conduct interviews, and interviewees are always requested to support their statements with evidence. A preliminary presentation of

assessment results completes the Execution activity. This is a review session aimed at correcting any misunderstanding that may have arisen when collecting information.

Hold opening briefing includes presentation of the PROFES team, what will happen in the assessment, as well as the expected results and benefits of process assessment. The opening briefing has two main goals:

- To inform the organization about process activities
- To strengthen the commitment of all relevant parties.

The purpose, content, and time schedule of the activities are presented, as well as who will be interviewed and when, and how the results of the assessment will be communicated, bearing confidentiality in mind.

An opening briefing lasts from 60 to 90 minutes, but can be reduced to a half-hour session in a re-assessment situation, for example. The audience should include all the affected participants from the SPU and the development projects.

Collect findings through interviews includes SPU and project level interviews. For each interviews there should be time allocated for defining the focus of the interview, i.e. what kind of information is specifically required from the interviewee. The focus is based on the studied material and earlier interviews. A single interview should not take more than two hours. Note that roles may also be combined, requiring fewer persons than mentioned here. However, there should be at least two persons in the PROFES team.

The elementary data extracted during the documentation analysis and interviews is recorded and scored according to the assessment methodology. Ideally, the assessors carry out the scoring immediately after each interview or documentation analysis meeting.

The result can be reviewed at SPU level before the project level interviews commence. The main purpose is to present the results to SPU interviewees, collect feedback, verify findings, and make corrections, if and when required.

For each project level interview, there should be time allocated for defining the focus of the interview, i.e. what kind of information is specifically required from the interviewee. The focus is based on the studied material and previous interviews.

The scoring sessions are performed after each interview as in SPU-level assessment. During the sessions, assessors analyse the information

collected in documentation analysis and interviews, write down the findings, and score practices and processes based on the findings.

In **Assess process capabilities**, the assessment results are rated, documented and verified, and assessment profiles are made. The results are reviewed jointly by the interviewees and interviewers in order to collect feedback, verify findings, and make corrections if necessary.

The results can also be used as a basis for defining assessment-related metrics for continuous assessment. Particularly useful are the assessment findings related to individual practices that can be traced to detailed indicators and tools, as these could be used during project implementation for a quick but effective assessment of progress. See Chapter 4 for more information on continuous assessment.

Activity 4-3

Reporting

- **Analyse results and derive improvement recommendations**
- **Prepare assessment report**
- **Deliver final results**

In this activity, the verified assessment results are analysed to identify suggested process improvement areas and priorities. The consolidated assessment results and improvement recommendations are documented and presented to the assessed organization to be used in subsequent PROFES steps.

In **Analyse results and derive improvement recommendations**, the verified assessment results are analysed. An analysis of strengths and weaknesses is performed based on the assessment results, which may also be compared to the possible benchmarking information obtained from the assessment methodology or measurement database of the target company. The detailed assessment findings are also analysed. As the result of the analysis, specific improvement actions are recognized and prioritized according to the business goals and improvement needs of the company.

In **Prepare assessment report**, all assessment results, strengths and weaknesses, and action plan recommendations are integrated into the assessment report, which is the final output of the assessment. The assessment report template defined in the assessment methodology can be used as a basis for this document. There are two different types of

assessment reports: SPU reports and project reports. The SPU report presents the findings of assessed SPU and summary results of the projects assessed. The project report is focused on the results of a single project, and is quite often only distributed to the project members. The preparation of the assessment report is performed as a two-step activity. A draft version of the report is first prepared and distributed for company-level review. The feedback is analysed and the final version of the report is then prepared.

Deliver final results includes delivery of the final on-site meeting results, the final version of the assessment report. The purpose of the final on-site meeting is to present the assessment findings to the whole site, based on the provisional final versions of the assessment report. It should be noted that presentations must respect confidentiality agreements signed during the PROFES process. The main topic of the event is to present SPU-wide strengths and weaknesses and improvement recommendations. The audience at the presentation should at least include management representative(s) and both SPU and project level personnel. Discussions and questions presented at the meeting will be taken into account when finalizing the assessment report to be delivered.

Average Duration of the Step

The total duration for analysing and evaluating current software processes will be about 4 weeks. The duration and effort needed depends on the number of projects assessed, and the size, complexity and maturity of the target organization.

Tools and Templates

- Presentations on PROFES for motivating management people (Management briefing)
- Presentations on PROFES for motivating project members (PROFES briefing)
- Commercial process modelling tools or simple graphic tools like ABC-Flowcharter can be used to produce the appropriate process models.
- We recommend that basic assessment tools be used to collect findings, to record assessment scores, and to count and present the ratings. The tools help assessors and make the assessment process more efficient. The more manual operations are needed, the slower and more laborious the assessment becomes. This increases the time gap between the interviews and feedback sessions. The assessment methodologies provide assessment tools, for example, BOOTSTRAP methodology includes the Bootsample tool for assessors.

- Assessment planning and reporting templates of the assessment methodology.

Work Products

Input work products	Output work products
<ul style="list-style-type: none"> • Business goals • Process descriptions • Quality manuals • Organizational characteristics 	Organizational level: <ul style="list-style-type: none"> • Process assessment reports and profiles • Descriptive Process Models • Preliminary improvement plan
<ul style="list-style-type: none"> • Project plan and other mgmt documents • Design documents • Measurement data 	Project level: <ul style="list-style-type: none"> • Process assessment reports and profiles • Descriptive Process Models • Preliminary improvement plan

Resource Allocation

Roles, responsibilities, and requested skills

Managerial roles

None specifically, but most likely many managers are interviewed in particular during SPU assessment.

Expert roles

Process modelling expert

Process modelling experts are responsible for carrying out the process modelling activities in Step 4. Process modellers can be external process modelling experts or the organization's quality engineer, for example. Process modellers recognize and describe current software process models with the help of interviewees. In some cases, a separate process modelling expert is not required, if a simple flow chart type process description is sufficient for assessment purposes. In this case, assessors can outline the process model during the assessment interviews.

Lead assessor

Lead assessor is qualified by an independent organization, and is responsible for the assessment. The lead assessor may be a person working internally in the organization or from an external organization, such as a consultant.

Assessor

Assessors are trained in basic knowledge and skills to participate in all assessment activities, including scoring and assessment report preparation. Assessors work together and under the guidance of the Lead assessor. Their role is to provide second opinions during evidence recording, scoring, and improvement planning. This helps to keep the assessment free of personal bias. Assessors may be external or internal in relation to the target SPU.

Interviewees

Interviewees are key persons within the organization. Typically these include top management representatives, product managers, quality managers, project managers in the assessed project, key software designers, etc.

Support roles*Facilitator*

Internal people responsible for arranging the assessments, interviews, required material, infrastructure, etc.

Expected effort/role*Process modelling expert*

The effort spent by the process modelling experts on the above activities will be up to 1 week each.

Lead assessor

The lead assessor will spend about 2 weeks on this step.

Assessor

Assessors will spend about 2 weeks each on this step.

Facilitator

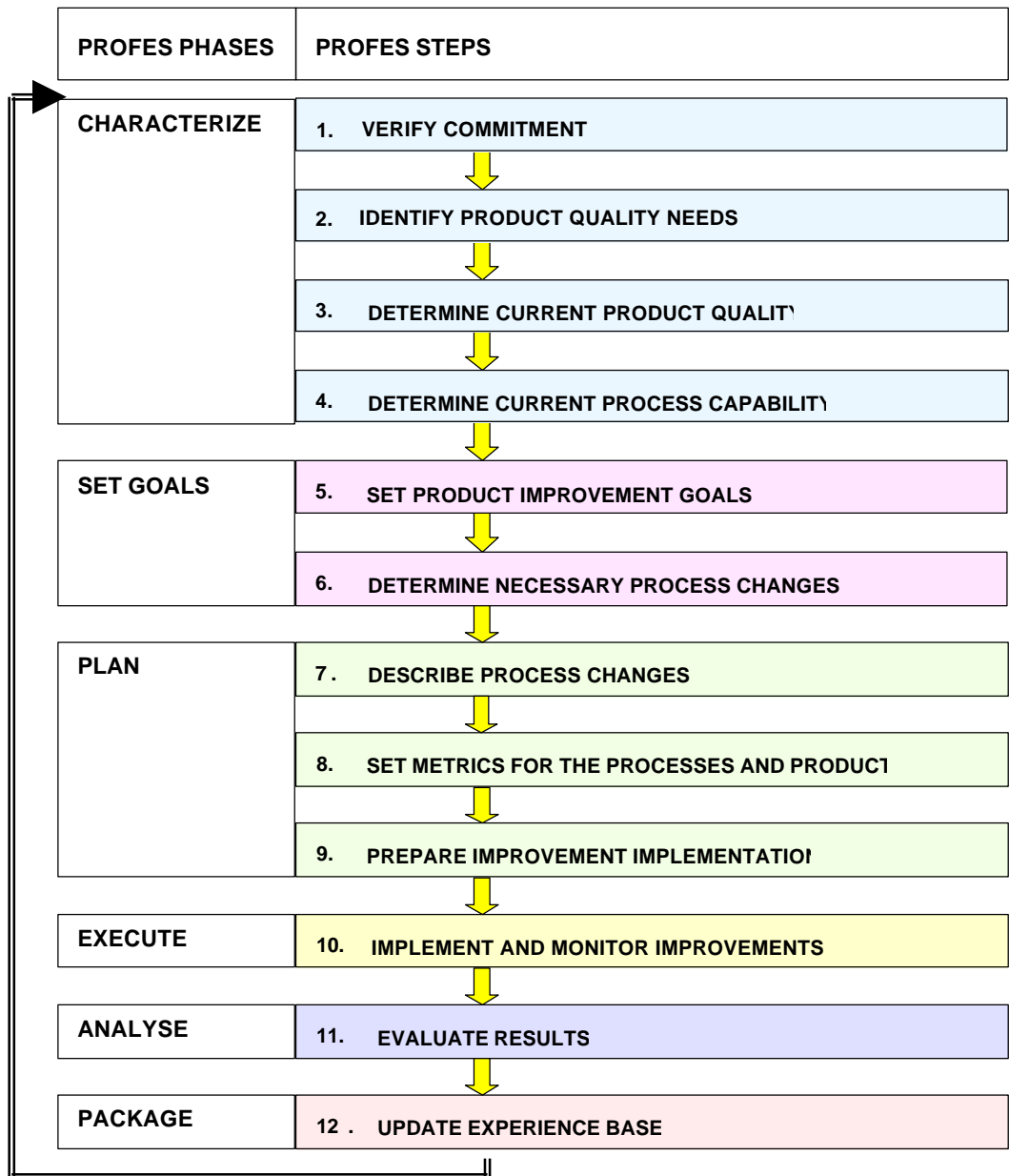
The effort spent by facilitators on the above activities will be about 1 week each.

Interviewees

The effort spent by interviewee on the above activities will be less than 8 hours each.

Methods and Techniques

- P. Kuvaja, J. Similä, L. Krzanik, A. Bicego, S. Saukkonen, G. Koch. 'Software Process Assessment & Improvement – The BOOTSTRAP Approach', Blackwell Publishers, 1994.
- BOOTSTRAP methodology including templates, tools, and data base for assessment preparation, implementation, reporting, and benchmarking (available through a user licence from BOOTSTRAP Institute)
- Rini van Solingen and Egon Berghout, "The goal/question/metric method, a practical method for quality improvement of software development", McGraw-Hill ISBN 007-709553-7, 1999.
- Process modelling languages, including ordinary natural language, structured natural language, template-oriented textual descriptions, flowcharts, activity diagrams, data flow diagrams, SADT diagrams, etc.
- ISO/IEC TR 15504-2: 'Information Technology - Software Process Assessment - Part 2: A Reference Model for Processes and Process Capability'. Technical Report type 2, International Organisation for Standardisation (Ed.), Case Postale 56, CH-1211 Geneva, Switzerland, 1998.



STEP**5****Set product improvement goals****PROFES****5-1 Analyse product quality discrepancies****5-2 Identify product improvement areas****5-3 Prioritize product improvement areas****5-4 Select the product improvement goals****Purpose**

The identification and installation of product improvement goals, based on product quality needs, current product quality, and current process status.

Step goals

- Product improvement goals are set.

Activities

The activities carried out during the setting of product improvement goals are highly dependent on the results of the preceding steps. Before this step can be carried out, we need to know:

- What are the product quality needs? We need to know what the product quality targets are, and which product quality characteristics are more important than others?
- What the current product quality is, in order to compare the ideal situation with the current one, and to determine where improvements are necessary
- What the status of the current process is? In order to identify where to improve the current process for improving product quality, we need to know what the status of the current process is. Based on the status of the current process, it becomes possible to identify in which area the process is more able to contribute to the target product quality

This input is necessary to determine the gap between the product quality required, and the current situation, and to determine the process improvement areas. However, the selection of process improvements is **not** included in this step, but postponed until Step 6.

The following activities are included in Step 5:

- Analyse product quality discrepancies

- Identify product improvement areas
- Prioritize product improvement areas
- Set the product improvement goals

**Activity
5-1*****Analyse product quality discrepancies***

- **Check the product quality needs**
- **Check the current product quality**
- **Determine discrepancies and record them**

Those product quality characteristics that are not fully satisfactory will be analysed during this step. This is done together with the results of previous steps, namely:

- Desired product quality, which describe the target quality levels of the product
- Current status of product quality, which describes the current quality of the product (if any), or which describes the product quality that is expected from the current process

During Step 3, these two sources of information have already been used to identify the preliminary product improvement goals. In this activity, the precise significance of these discrepancies is analysed.

The implications and causes of the discrepancies are analysed. The gap between current and target quality is of especial interest, since it points to the product improvement areas. Where necessary, corrective action is taken by means of process changes. Such information on discrepancy is valuable at this point of time, as there is still time to make alterations.

There are two possible reasons for discrepancies:

- Positive – current product quality is already higher than necessary.
- Negative – current product quality is lower than required, so if no action is taken, the target will not be reached.

The best thing to do is to list the ISO9126-quality sub-characteristics and identify what the target and current status is for each of them. Based on these findings, differences can already be pointed out. This same task can be repeated for the quality characteristics, however input from the sub-characteristics is also now available for application. PROFES improvement methodology does not prescribe a scale on which quality can be expressed, but we recommend using a quantitative scale where possible. Should this not be possible, qualitative expressions such as (high, medium, low) can be used.

**Activity
5-2*****Identify product improvement areas***

- **Check the product quality discrepancies**
- **Select those product quality sub-characteristics that need improvement**
- **Report the areas of product improvement**

Once the product quality needs are known and specified in generic terms such as ISO9126, and the current product quality is known in the same terms, product improvement areas can be identified. These product improvement areas can also be identified, based on the results of Step 4 (Determine current process capability), in case the process assessment has identified product improvement areas while studying the process.

Differences between target product quality and current product quality, especially those situations where the current level is lower than the target level, indicate the possibilities for improvement.

Therefore, it is best to start by making a list of all ISO 9216 product quality characteristics and sub-characteristics, and checking each item for any discrepancies and how large they are.

Based on such an overview, analysis and discussion can be held by or with the project manager on what is identified and which specific product quality (sub-) characteristics need improvement within the project's terms of reference. These are the product improvement areas on which the process improvement programme could focus. No decision should be taken yet.

The aim of this activity is to identify the product improvement areas, and to clarify the rationale behind each possible product improvement action, in order to support decision-making on product improvement goals.

Activity 5-3	<i>Prioritize product improvement areas</i>
-------------------------	--

- **Analyse the product improvement areas**
- **Analyse product quality needs and their relation to business results**
- **Assign priorities to each improvement area**

Before the final product quality goals can be set, all product improvement area identified in the previous activity should be prioritized. This can be based on a discrepancy factor, i.e. the larger the discrepancy between target and current quality, the higher the priority, but also on the relation between the product improvement area and its impact on business results.

Priority determination is a difficult process in which many parties should be involved, or at least consulted. Marketing, management and customers in particular should be heard when setting the priorities. Therefore, decision makers from several parties need to be involved in this activity.

Product and process improvements are still an investment, and those investments that will have the largest (or quickest) payback should be favoured.

Decisions on product quality goals can be taken, based on these priorities. Such decisions are very important but are not often easy to make. However, the previous PROFES steps have provided much information and support for making these decisions.

**Activity
5-4*****Select the product improvement goals***

- **Select the product improvement areas that have the highest priority**
- **Analyse the feasibility of each product improvement area**
- **Establish the product improvement goals**
- **Achieve management commitment**

The discrepancies identified were prioritized in the previous activity. In practice, not all the problem areas are likely to be tackled and so they must be selected in some way. We recommend discussion or brainstorming techniques to help carry out this process, which should at least involve the project manager or other responsible person. Based on this prioritized list of discrepancies, product improvement goals can then be selected.

We recommend selecting the product improvement goals together with the complete project team, to make sure that the project team supports the product improvement goals. Motivation for product improvement is very important, and so the project team must be involved as much as possible. Furthermore, product improvements should be checked for feasibility. Although certain product characteristics might have a high improvement priority, improvement will not always be possible within real-world constraints. In such cases, improvement efforts can best be focused on other areas where results can be achieved with less effort.

Product improvement goals can be specified simply by listing the product quality (sub-) characteristics to be improved. However, we recommend that product quality goals be specified in measurable terms. For example, if maturity (reliability) is an improvement area, we recommend specifying this goal in terms of mean time between failures, number of field defects, down time, etc. A famous quote by Tom Gilb reads: "Projects without clear goals will not achieve their goals clearly!" This supports the idea that the more concrete and measurable improvement goals are, the better people become at achieving them.

The results of this analysis and decision process should be documented. Not only the final decision is important, but also the rationale behind that decision. In the future, it will be necessary to clarify the objectives and the rationale behind them, in order to prevent incorrect changes to the improvement objectives. It is very important to document why certain product improvement areas are selected and others are not.

Management is essential for any process or product improvement programme, and complete management commitment is therefore recommended. This will help to facilitate the implementation of process changes in future PROFES improvement methodology steps.

Average Duration and Effort

Average duration of the step is 2-3 calendar weeks, largely depending on the availability of people during that period. The total effort is about 40-60 person hours.

Tools and Templates

Beside the use of some templates, PROFES does not prescribe the application of any particular tool.

Work Products

Input work products	Output work products
<ul style="list-style-type: none"> • Business goals • Product quality needs • Product quality target profile • Current status of product quality • Process assessment reports and profiles • Preliminary product quality goals • Product characteristics 	<ul style="list-style-type: none"> • Product improvement goals

Resources Allocation

Roles, responsibilities and requested skills

Managerial roles

PROFES Team Manager

The PROFES team manager is involved in prioritizing problem areas of product quality, and will be involved during the selection of product improvement goals. The PROFES team manager will also review the PROFES experts' deliverables and other work, and will communicate the results of this step to the project manager and project team.

Project Manager

The project manager is involved during the prioritization of discrepancies, and is mainly responsible for the selection of product improvement goals. The project manager is also responsible for ensuring that the project team supports the product improvement goals and is motivated towards attaining them.

Decision-maker

Determining the product improvement areas is a step in which decision-makers need to be involved. They should be consulted especially during the definition of priorities for each possible improvement area. Decision-makers are people from marketing, management, or customers that have a strong voice in decisions regarding the product, e.g. sales, acquisition, and evaluation.

Expert roles

PROFES expert

The PROFES expert will facilitate most of the work in this step, and is responsible for the production of all deliverables. As this step involves some quite difficult tasks, the PROFES expert should have a thorough knowledge of software processes and product quality.

Support roles

Project team

The project team will be involved in reviewing the results of this step, and in the selection of the product improvement goals. It must completely support the product improvement goals, and so its involvement is crucial.

Expected effort/role*PROFES expert*

The PROFES expert produces most of the deliverables, requiring about 16 hours of effort.

PROFES Team Manager

The PROFES team manager is mainly involved in the meetings and review of documents, about 10 hours of effort

Project Manager

The project manager is mainly involved in the meetings and review of documents, about 10 hours of effort

Project team

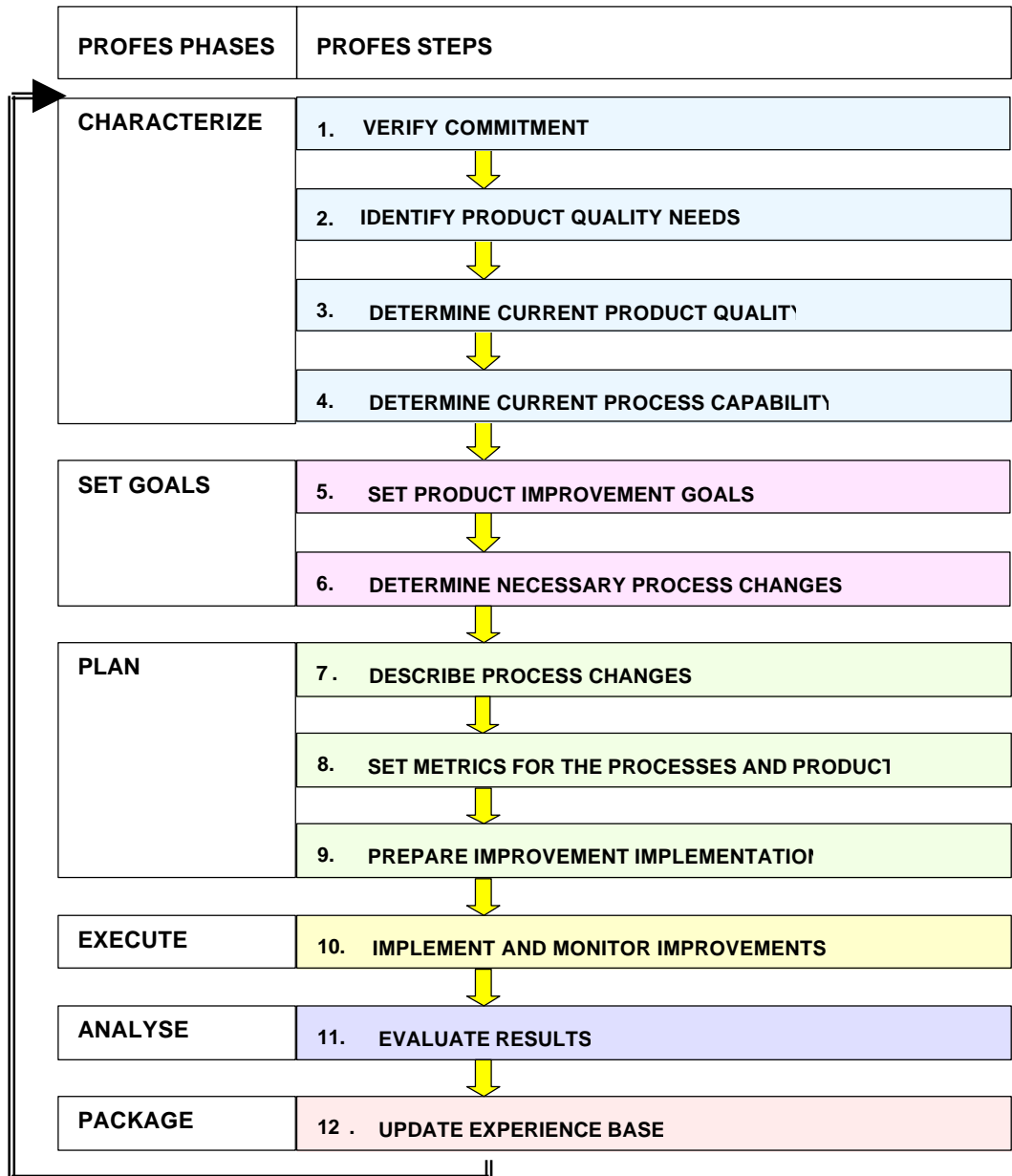
The project team attends a meeting for prioritizing problem areas and selection of product improvement goals, about two hours per project team member.

Decision-maker

The effort spend by the decision-makers on this step is less than 4 hours each.

Methods and Techniques

No methods are prescribed. However, techniques such as group brainstorming or multi-criteria decision methods can be useful.



STEP 6	Determine necessary process changes	PROFES
	6-1 Identify product quality goal 6-2 Identify processes to be improved 6-3 Retrieve relevant PPD models 6-4 Construct characterization questionnaire 6-5 Characterize the project 6-6 Rank PPD models 6-7 Select improvement actions	

Purpose

The purpose of this step is to identify and select process changes that contribute to the achievement of product quality goals. This is carried out using PPDs, product improvement goals, and process status.

An organized approach to the identification of process changes using PPDs is necessary in order to benefit from existing experience of effective improvement actions. It leads to well-informed decision-making and reduces the risk of improvement programme failure. Furthermore, the use of PPDs makes the decision process explicit and transparent. This facilitates later evaluation of the improvement programme and allows for updating the PPD repository according to the evaluation results.

Step Goals

- Identify and select process changes to achieve the product improvement goals
- Document the decisions on necessary process changes for later evaluation of the improvement programme

Activities

The step is divided into the following activities:

- Identify product quality goal
- Identify processes to be improved
- Retrieve relevant PPD models

- Construct characterization questionnaire
- Characterize the project
- Rank PPD models
- Select improvement actions

**Activity
6-1*****Identify product quality goal***

- **Review the product improvement goals identified previously**
- **Prioritize product improvement goals**

In goal-driven, product-focused process improvement, the identification of product improvement goals is essentially the first step. The early steps of the PROFES improvement methodology provide guidance for identifying such product quality goals: Identify product quality needs (Step 2), identify current process capability (Step 4), and set product improvement goals (Step 5). As a starting point for the selection of improvement actions, the goals identified in these previous steps should be briefly reviewed. We recommend addressing only a few product quality goals at the same time. Therefore, it might be appropriate to select one single product improvement goal first, and address other possible goals during later stages of the improvement programme.

The following activities for identifying improvement actions are illustrated using an example case, for which the product improvement goal for the software part of a newly developed embedded system is:

“Improve product reliability by avoiding any severe field defects during the first six months of operation.”

For reasons of simplicity, there is only one product improvement goal being used in the example case. However, the steps would apply equally well to two or more product improvement goals.

**Activity
6-2*****Identify processes to be improved***

- **Use PPD repository for identifying processes with greatest effect on product quality**
- **Select processes for which improvement actions are both important and feasible**

Once the product quality goal is determined, the next step is to identify the processes to be improved. This is the first point in time when explicit information about product/process dependencies can be used, as it is stored in a PPD repository. The identification of the processes to be improved can be divided into three independent questions:

1. Which processes are expected to have the greatest effect on the required product quality?
2. Which processes have highest improvement potential?
3. Which processes can actually be modified?

Answering each of these questions individually provides a comprehensive picture of those prospective processes in which appropriate improvement actions can be performed. The PPD repository helps to identify those processes with the greatest effect on the product quality required. Therefore, the processes contained in PPD models for the quality attributes of product improvement goals should be queried. The web-based PROFES PPD repository contains an index for accessing those processes via HTML links. In an example case, such a query provided the following six processes: Software requirement analysis, software architecture design, software detailed design, software implementation and testing, software integration and testing, and lifecycle methodology. This means that each such process can be expected to be particularly relevant for achieving high product reliability. The evidence for this comes from past project experience that has been incorporated into the PPD repository. It should have been thoroughly validated, with additional information accessible through the repository, thus allowing the decision-maker to assess the relevance of this evidence in the given project case.

The improvement potential of a process can be viewed analogously to its software development capability level, as it can be measured for instance with ISO/IEC 15504 process assessments. The result is a profile of the

current capability levels. Processes with low capability levels usually have the highest improvement potential. This means that there are many prospective practices and technologies available by which the manageability and performance of these processes can be further improved. Naturally, other approaches are also possible for identifying those processes with the highest improvement potential. Often, project managers and personnel know the specific strengths and weaknesses of their development processes quite well and can come up with some good improvement measures. The advantage of process assessments, measurement programmes, quality measurement assessments and the like is that they can uncover process-related aspects that were not yet known to a software organization. In our example case, the following processes turned out to have the highest improvement potential: Software requirement analysis, software implementation and testing, software integration and testing, configuration management, and risk management.

Usually, not all prospective processes for improvement actions can be changed. Possible reasons are lack of resources, recent changes of processes that have not yet stabilized sufficiently, expected reluctance of the personnel, etc. In our example case, the manager decided not to change software implementation and testing, as the project schedule was tight and any change in later phases that affected larger parts of the project was expected to put in-time delivery at risk. It was likely that developers would resume their old implementation and testing practices, should time pressures occur. Changes to software integration and testing were not seen as harmful, as it was the task of a highly motivated experienced integration engineer to support improved integration practices.

Table 3.2 shows the three sets of prospective processes and highlights those that are contained in every category. These are the processes for which improvement actions are sought. The project manager expects the required product quality goal (i.e. high reliability) more likely to be fulfilled with appropriate improvement actions for one or more of these processes, rather than with the current, unchanged development process.

Table 3.2. Selection of prospective improvement actions.

<i>Processes with particular product quality impact</i>	<i>Processes with highest improvement potential</i>	<i>Processes that can actually be modified</i>
Software requirements analysis	Software requirements analysis	Software requirements analysis
Software architecture design		
Software detailed design		
Software implementation and testing	Software implementation and testing	
Software integration and testing	Software integration and testing	Software integration and testing
Lifecycle methodology		
	Configuration management	Configuration management
	Risk management	Risk management

**Activity
6-3**
Retrieve relevant PPD models

- **Query the PPD repository for PPDs that link the product quality goal to the processes to be improved**
- **Build a collection of prospective improvement actions from these PPDs**

Based on the previously identified product quality goal and the prospective processes for changes, the PPD repository can be queried again, this time for all PPD models that refer to the selected product quality attribute and processes. In our example case, such a query has resulted in PPDs for the following technologies:

- Software inspections for requirements documents

- Software inspections for architecture and detailed design documents
- Cleanroom software engineering principles for detailed design and system testing
- Personal software process for module-level development (focus here on detailed design)
- Incremental integration technique for integration and testing

Each such technology is associated with one or more PPD models of the type shown in Table 3.3.

Table 3.3. Example of PPD model

PPD Model	
Product Quality	Reliability
Process	Software Requirements Analysis
Technology	Software Inspections
Context	
Size of inspection team	1-2 3-5 6-8 9-10
Experience of inspection team	low average high
Problem treatment of inspection team	pragmatic detailed
Complexity of inspected document	low average high very_high
Size of inspected document	small average large very_large
Management commitment	low high
Overall time pressure	low average high
Module affected by new hardware	old_hw new_hw
Module developed externally	internally externally

The retrieved technologies represent the prospective improvement actions for the project. In principle, each of them can be applied for one or more of the processes that are candidates for process changes, and all can be expected to contribute significantly to further improvements of product reliability. However, it can be expected that not all prospective process changes are equally relevant for this project, because some might not work very effectively in the particular context of this project. Others might require more time for a proper technology introduction than is available under the given constraints. Hence, the technologies should be briefly analysed with regard to their fit to the project's specific characteristics. Only those well suited to the given project should be selected, and which can be introduced without any severe risks of failure.

Whether a paper-based document, an on-line database, or a web repository with integrated decision support processes, different techniques are available for finding the most suitable individual technology for the project. They depend on the way a PPD repository is implemented, and on the number of prospective improvement actions. In any event, the context information contained in the PPD models provides the basis for identifying the degree of suitability with the project. The following steps describe the basic principles of the comparison. They can be implemented and supported by tools in various ways.

**Activity
6-4**
Construct characterization questionnaire

- **Collect the context factors that affect the selected improvement actions**
- **Build a characterization questionnaire from these context factors**

How well a prospective improvement action suits a project must be decided based on what is known about the improvement action's context requirements. In a PPD model (see Table 3.3), such requirements are described in the form of context characteristics (i.e. the lower part of a PPD model). The project must be characterized with regard to these characteristics. Then each prospective technology's PPDs can be compared to the project characteristics, so that the goodness of fit between improvement action and project can be determined.

A characterization questionnaire for a project can be derived from the context factors. They are part of the PPD models for the prospective improvement actions, such as technologies that may possibly be introduced. All the context factors of these PPD models must be collected, possible duplicates should be removed, and a questionnaire should be constructed out of them. In our example case, the PPD model for requirements inspections is the one shown in Table 3.3, and other prospective improvement actions contribute context factors like those shown in Table 3.4. This table shows the characterization questionnaire for the example case. For instance, the context factor "project team organized in small sub-teams according to system components", belongs to the PPD model of Cleanroom software engineering. The context factor "PSP training for the whole team done or possible" belongs to the PPD model of PSP for module development.

Table 3.4. Filled-out characterization questionnaire

Management commitment for inspections?	low high
Overall time pressure of project?	low average high
Modules affected by new hardware?	old_hw new_hw
Modules developed externally?	internally externally
Project team organized in small sub-teams according to system components?	Y N
PSP training for the whole team done or possible?	Y N
Time allocation for early project phases?	low high
Measurement of defects introduction and identification per phase?	Y N

**Activity
6-5**
Characterize the project

- **Describe the characteristics of the project**
- **Use the previously constructed characterization questionnaire**

Given an appropriate characterization questionnaire, the project can be easily characterized with regard to its characteristics that are relevant for the selection of improvement actions. Table 3.4 shows the project characterization of the example case. The project manager has left the question about time allocation for early project phases unanswered, because she would like to spend more time on specification and design but she is not yet sure whether this can be afforded. She did not want to exclude any prospective improvement action based on this still-undecided aspect. The project manager could also have written two alternative characterizations, and check later on whether they lead to different recommended improvement actions.

**Activity
6-6****Rank PPD models**

- **Compare the context models of the prospective improvement actions with the context characterization of the project**
- **Rank the prospective improvement actions with regard to their goodness of fit with the project characteristics**

Based on the project characterization provided in the previous step, each prospective improvement action can be checked for how suitable it is for the project. The respective PPD models can be ranked according to their goodness of fit. Those that have the strongest relative overlap with the actual project characteristics should be ranked highest. For those PPD context characteristics that deviate from the project characteristics, it should be checked whether the deviations are really critical. In other words, can they impose severe risk on successful implementation of the improvement action? If this is the case and the risk can not be mitigated well, then the respective improvement action should be dropped.

Table 3.5 shows the following ranking of improvement actions and their related processes as occurred in our example case. Various types of software inspections have achieved the highest ranking, followed by the Incremental integration technique, Cleanroom, and the Personal Software Process (PSP). Cleanroom and PSP have ranked lowest, mainly because they are quite comprehensive approaches that have specific requirements for quite many aspects of the software development process. Their introduction also requires considerable time and resources for training and trial use. These resources were not made available to the project.

Table 3.5. Ranking list of prospective improvement actions.

<i>Rank</i>	<i>Technology</i>	<i>Process</i>
1	Software inspections	Software requirements analysis
1	Software inspections	Software architecture design
1	Software inspections	Software detailed design
2	Incremental integration technique	Software integration and testing
3	Cleanroom software engineering	Software detailed design
3	Cleanroom software engineering	Software integration and testing
3	Personal software process for module-level development	Software detailed design

**Activity
6-7****Select Improvement Actions**

- **Review the ranking list of prospective improvement actions**
- **Select the actual improvement actions for the project**

The ranking of improvement actions resulting from the previous step assists the decision-maker through the final selection of project improvement actions. The ranking list, project characterization, and the various kinds of information associated with the PPD models provide a good basis for well-informed decision making. In particular, the following detailed information sources are available from the previous steps of the decision support process and the associated PPD repository:

- An explicit characterization of the project and its most important product quality goals
- A list of processes that are most important or critical for the achievement of the product quality goals
- A list of prospective improvement actions that support the achievement of the product quality goals
- A ranking list of improvement actions with regard to their goodness of fit for the project characteristics
- Detailed information about the deviations between the context factors of the most appropriate improvement actions, and the project characteristics
- Definitions of the improvement actions, i.e. software engineering technologies that allow the decision-maker a quick and detailed understanding of what the improvement actions imply for the project
- Links to past projects in which the improvement actions were implemented, together with experience about the success of the improvement actions in these projects.

Based on this information, software project or improvement programme planners can gain knowledge that they were possibly not aware of before. The systematic decision process fosters important aspects of the project and makes the improvement programme explicit. This helps to identify and mitigate potential project risks. It also helps to secure the commitment of higher management for the implementation of the improvement actions.

In our example case, the project manager has selected software inspections for requirements, architectural design, and detailed design documents, as well as the technique for iterative integration. These have turned out to be suitable for the project characteristics. They can also be implemented with the resources available for the new project. Nevertheless, during the decision process, the project manager got to know the Cleanroom and PSP better. She decided to prepare for the introduction of statistical usage testing (a Cleanroom technique) in another project that provided more time for preparing the necessary changes and training efforts. The information on PSP sparked an initiative for bringing quality assurance closer to the individual engineers. A quality manager started to investigate how the developers could receive more direct feedback about the quality impact of their individual work practices. As a result, the accessibility of the corporate measurement database was improved, and PSP training was offered to the engineers.

Average Duration and Effort

The duration and effort for the selection of improvement actions depend on the number of improvement goals and complexity of the organizational context. Complex organizational contexts usually require more effort for checking whether a prospective improvement action is suitable. An overall effort of 1 to 3 days can be expected.

Tools and Templates

Tools:

- The web-based PROFES PPD Repository:
<http://www.iese.fhg.de/Profes>
- Andreas Birk, Felix Kröschel. "A knowledge management lifecycle for experience packages on software engineering technologies", in Proceedings of the Workshop on Learning Software Organizations, Kaiserslautern, Germany, June 1999.

Templates:

- PPD template for process-impact PPDs (see Appendix 2)
- PPD template for context-impact PPDs (see Appendix 2)

Work Products

Input work products	Output work products
<ul style="list-style-type: none"> • Product improvement goals • Process assessment reports and profiles (result from Step 4) • PPD repository (see Section 7) • Preliminary improvement plan (result from Step 4) 	<ul style="list-style-type: none"> • Process changes to be implemented in the improvement programme • Characterization of the project or improvement programme

Resource allocation

Roles, responsibilities and requested skills

Expert roles

Experience Base Supporter

Mainly acts as decision facilitator in a role that supports the project based on information collected in the experience base.

Managerial roles

Project planner

Characterize project or improvement programme, review prospective improvement actions, and select improvement actions to be implemented

Expected effort/role

Expert role: Experience Base Supporter

The typical effort of the Experience Base Supporter is one to two days. Most of the effort during the selection of improvement actions is for the EB Supporter role, which acts as a decision facilitator.

Managerial role: Project planner

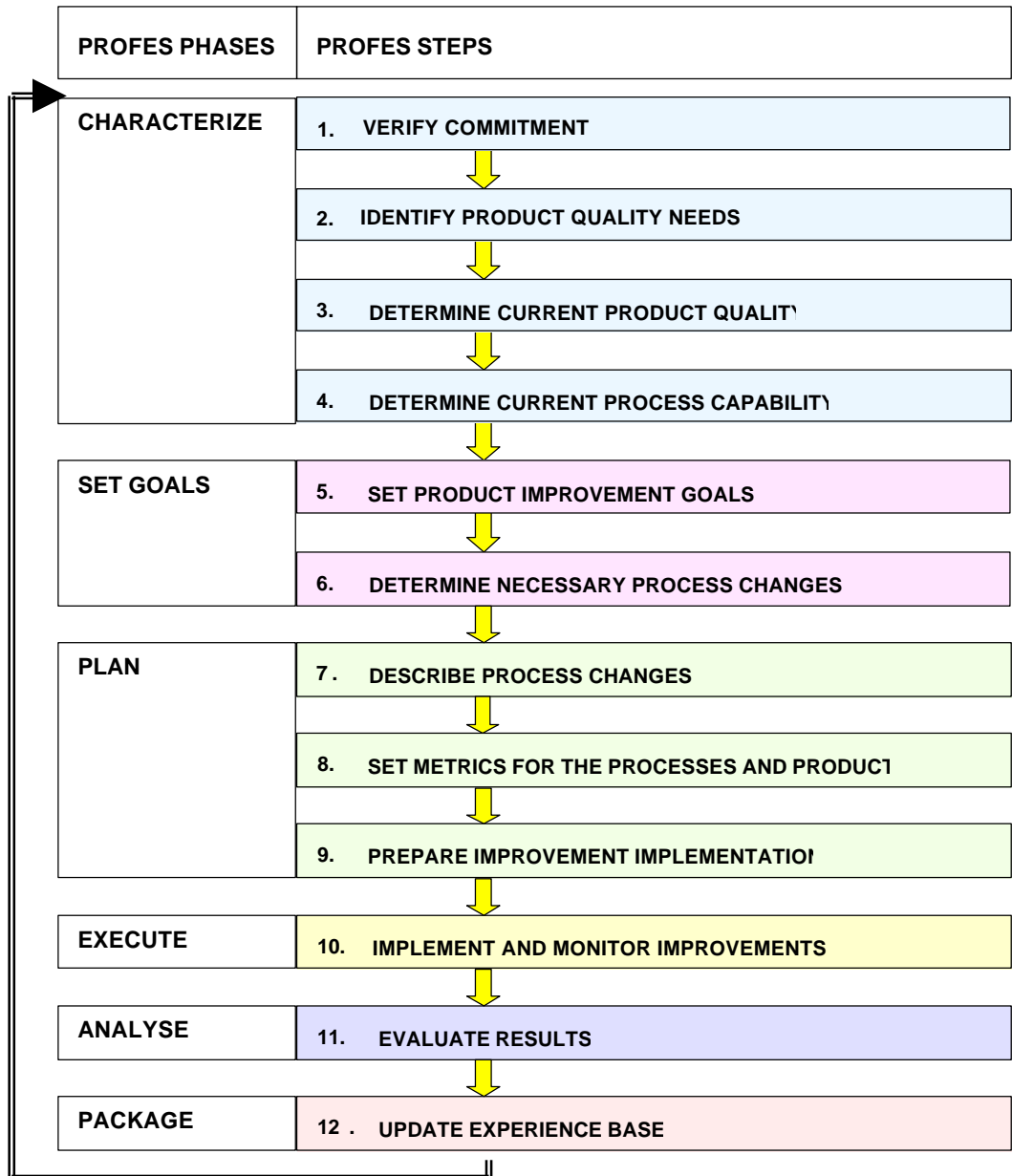
The typical effort of the project planner for the selection of improvement actions also ranges from one to two days. Usually, project planners have to invest less effort than EB supporters.

Methods and Techniques

So far, only little systematic support has been developed for the selection of improvement actions. Two basic strategies are (1) Similarity-based knowledge retrieval and decision support; this requires that a PPD repository is available, and (2) Multi-criteria decision-making methods; these do not usually require an elaborated PPD repository.

Similarity-based knowledge retrieval and decision support: Klaus-Dieter Althoff, Andreas Birk, Christiane Gresse von Wangenheim, and Carsten Tautz. "Case-Based Reasoning for Experimental Software Engineering". In M. Lenz, B. Barsch-Spörl, H.-D. Burkhard, and S. Wess, editors: "Case-Based Reasoning Technology – From Foundations to Applications", pages 235–254. Springer-Verlag, Berlin, 1998.

Multi-criteria decision-making: Mansooreh Mollaghasemi and Julia Pet-Edwards. "Making Multiple-Objective Decisions". IEEE Computer Society Press, November 1998.



STEP 7	Describe process changes	PROFES
	7-1 Mark processes/practices in the current process model, which have to be changed	
	7-2 Develop prescriptive process model	
	7-3 Communicate prescriptive model to participants in the process	

Purpose

Changes in the development process, as identified in Step 6 based on the use of PPD models, are worked into a prescriptive process model (see Glossary and Appendix 1) that is recommended for use in the following project. The prescriptive process model integrates the altered software engineering practices with the current development process, as identified in Step 4 in the form of a descriptive process model. Adaptation of the new practices will possibly be necessary in order to suit the final integrated process. The new prescriptive process model has to be communicated to the organization.

Step Goals

- Agree and document future performance of the development process
- Achieve clear understanding of relevant processes in order to define the metrics in the following step

Activities

The step is divided into the following activities:

- Mark processes/practices to be altered in the current process model. A completely new process may also need to be created.
- Develop prescriptive process model
- Communicate prescriptive model to process participants

Please note that the development of the prescriptive process is typically an iterative process requiring close co-operation between process owners, process modellers, and those software engineering people affected. The level of implementation depends greatly on the needs and context of the

organization. However, often a simple textual description of the processes is sufficient.

**Activity
7-1*****Mark processes/practices in the current process model, which have to be changed***

- **Locate those processes and practices in the current process model that have to be changed or adapted.**
- **Identify those processes and practices that have to be deleted.**
- **Identify locations in the current process model where new processes/practices will be included.**

Based on the output from Step 6, the processes and/or practices that have to be changed, replaced, or adapted in the existing process model are located.

To understand the relationships and effect of the selected process changes, information from project post-mortem analyses, final reports, test- and quality reports of previous projects, or specific ‘opportunities for improvements’ meetings can be used.

**Activity
7-2*****Develop prescriptive process model***

- **Update current process model according to marked changes**
- **Establish prescriptive process model**
- **Test prescriptive process model**

The prescriptive process model is built by modifying the relevant practices/processes in the descriptive process model with the chosen process changes. This may include interviews, workshops, etc. to gain an adequate understanding of the identified process changes, and the necessary level of process documentation. We recommended that the new prescriptive process model be tested before generally adopting new and altered practices. Some things to remember are:

- The exact description of the modelling procedure depends on the process modelling technique and method chosen
- Process changes can also be adapted for a project by describing them in the project plans, or in separate process/method descriptions
- It is advisable that project personnel participate in the creation of the new prescriptive process model, to ensure its technical quality and the motivation of the project personnel.

As there are a multitude of methods, techniques, and tools available, it is beyond the scope of the PROFES User Manual to offer more precise guidelines for process modelling. The process modelling aids may also change from what they are at the moment. However, a prescriptive process model should address the following:

- Artifacts consumed, produced, and modified
- Activities carried out
- Agents (with roles) involved
- Tools and technologies (techniques, methods, etc.) used
- Entry and exit criteria associated with activities
- Relationships between activities and artifacts, i.e. flow of artifacts
- Assignment of roles to activities
- Application of tools and technologies (techniques, methods, etc.) in activities
- Relationships between products, i.e. product hierarchies
- Relationships between roles, i.e. communication network.

Please note that the prescriptive process model should contain not only an overall description of the process to be performed, but also give adequate practical guidance on **how** to perform the process. This information may be in the form of work instructions, templates, etc.

**Activity
7-3*****Communicate prescriptive model to process participants***

- **Distribute the prescriptive process model to the people concerned**
- **Train people to use the new prescriptive process model**

The newly developed Prescriptive Process Model (PPM) and related templates, examples, etc. have to be distributed to all people who will be using it directly or indirectly:

- In presentations for project personnel on different organizational levels, using workshops, demonstrations etc.
- Through process simulations and games
- Via Intranet
- Via in-house mail (on paper)

The capabilities of project personnel for using the new process model should be ensured. If people have participated in the development of the new process model, this should not prove difficult. Some training may nevertheless be necessary.

Average Step Duration and Effort

The effort can range between one day to a month, depending on the extent of the process changes, which should not be overly delayed.

Tools and Templates

The process modelling tool used by the company is sufficient. Other possible tools are:

- Word processors, MS-Word, etc.
- Simple graphic tools, ABC-Flowcharter, etc.
- Comprehensive tools, SPEARMINT, STATEMATE] etc.
- Presentation tools, Intranet for communications
- Web-based workflow and groupware tools

Work Products

Input work products	Output work products
<ul style="list-style-type: none"> • Descriptive process model (result from Step 4) • Selected list of process changes (result from Step 6) 	<ul style="list-style-type: none"> • Prescriptive process model (including selected process changes) • Training/presentation material for the new process

Resource Allocation

Roles and responsibilities

Managerial roles

Process owners

- § Control all process modelling activities
- § Follow the new process model and support its use

Project managers

- § Prepare to use the new process model, consider impact on planning
- § Present and distribute information

Quality assurance managers

- § Participate in process modelling, consider impact on quality activities.

Expert roles

Process modellers

- § Provide information on process changes
- § Implement process modelling activities

Project members

- § Participate in process modelling as technical experts
- § Use the new process model and report experience

Support roles

Process modelling support

- § Assists process modelling activities, e.g. documentation, organizing meetings, training, etc.

Expected effort/role

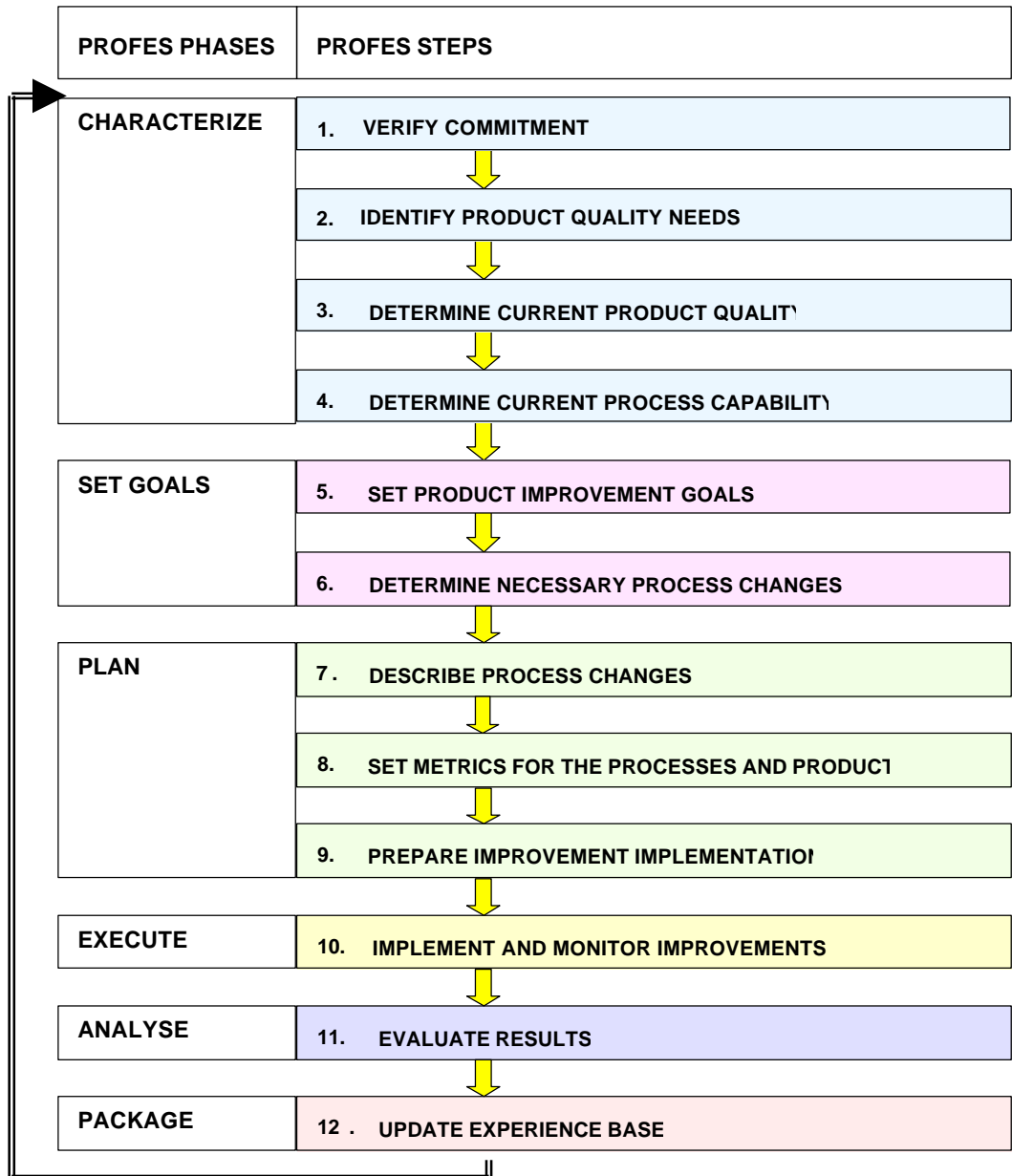
Depends on the extent of the implementation. Typically, the process owner and process modeller will make the most effort, and project members the least effort. Please note that project effort may be affected.

Methods and Techniques

Generally, the actual approach depends on the process modelling method and technique chosen. Company-specific instructions on how to make changes in process models should be applied. Initiatives for changes can be communicated via e-mail to the organization responsible for development of the process model.

There are many possible methods and techniques for process modelling. Some of the better-known process modelling methods with at least some tool support are SPEARMINT, MVP-L, APEL, STATEMATE and OPSIS.

Finally, please note that process modelling does not necessarily require sophisticated methods, techniques or tools. Often a simple textual description of the processes offers a good basis for achieving an adequate understanding of the process to be followed. However, web-based workflow and groupware tools are increasingly used to describe and support processes.



STEP**8****Set metrics for the processes and product improvements**

PROFES

8-1 Define measurement goals**8-2 Conduct GQM interviews****8-3 Define questions and hypotheses****8-4 Define and check metrics****8-5 Produce GQM plan and measurement plan****Purpose**

This step is carried out to monitor and control the altered development process, evaluate the changes and the underlying PPD models, or demonstrate achievement of the overall improvement success. The collection and analysis of measurement data in the development process and products help to achieve the defined goals. Refinement of this goal to a measurable level is carried out according to the GQM paradigm (Goal/Question/Metric).

Step goals

- Define a set of questions and metrics related to the product quality goals
- Define a set of questions and metrics related to the process performance goals
- Define a set of questions and metrics related to the product-process dependency goals.
- Construct GQM plan and measurement plan

Activities

The step is divided into the following activities:

- Define measurement goals
- Conduct GQM interviews
- Define questions and hypotheses

- Define and check metrics
- Produce GQM plan and measurement plan

Please note that this step requires expertise of building measurement programmes.

**Activity
8-1*****Define measurement goals***

- **Consult project team members**
- **Specify measurement goals**
- **Prioritize measurement goals**
- **Select measurement goals**

The first step in the GQM definition phase is to define formal measurement goals. To do this, the PROFES team must consult all the people participating in the measurement programme. In principle, these people are probably software engineers, although some improvement goals might involve other people. All people collecting measurement data also need to be involved in measurement goal setting. Any possible measurement goals stated by these people should be made explicit, after which the goals can be prioritized and a final decision made on the measurement goals that are selected.

Measurement goals in the context of PROFES are based on the product quality goals, process conformance goals, or product-process dependency goals.

Measurement goals must be defined and described on such a level of abstraction that goal attainment can be evaluated by analysing data collected for the purpose. Therefore, measurement goals must be defined in a structural and logical way. For this purpose, templates are available that support the definition of measurement goals by specifying the purpose (what object and why?), perspective (what aspect and who?), and the environmental characteristics. The PROFES template for measurement goals is included in the PROFES GQM plan template in the appendices to this manual.

**Activity
8-2****Conduct GQM interviews**

- **Select project team members for interview**
- **Prepare interview**
- **Hold interview**
- **Report interview**

The project team must be closely involved in the development of the measurement programme. With respect to the defined measurement goals, structured interviews with individual members of the project team must be held to extract the knowledge from the project team

In Step 4, interviews were also conducted. The object of these interviews was to determine the current capability of the process, while the GQM interviews clarify the GQM questions and metrics. However during the PROFES project, in some cases it was possible to combine these interviews, as described in Chapter 4 of this manual.

So-called 'abstraction sheets' are used during the interviews. An abstraction sheet summarizes the main issues and goal dependence as described in the GQM plan and presents this information divided into four sections on a page:

Quality Focus: What are the measured properties of the object, according to the project members?

Baseline Hypothesis: What is the project members' current knowledge, with respect to these measured properties?

Variation Factors: Which (environmental) factors do the project members expect to have an effect on the measured properties?

Impact on Baseline Hypothesis: How do these variation factors influence the measured properties?

An example of an abstraction sheet is shown in Figure 3.4. The four sections can be checked for consistency and completeness. The four sections are mutually related. For example, for every quality focus,

hypothesis should be stated; or for every variation factor, its effect on the hypothesis should also be made explicit.

<u><i>Object</i></u>	<u><i>Purpose</i></u>	<u><i>Quality Focus</i></u>	<u><i>Viewpoint</i></u>
<i>Delivered Product</i>	<i>Understanding</i>	<i>Reliability and its causes</i>	<i>SW Development Team</i>
<u>Quality Focus</u>		<u>Variation Factors</u>	
Number of Failures		Level of Reviewing	
<ul style="list-style-type: none"> By Severity By Detection group Number of Faults By Module 			
<u>Baseline Hypothesis (estimates)</u>		<u>Impact on Baseline Hypothesis</u>	
Distribution of Failures		The higher the level of reviewing,	
By Severity		the less failures, and the less faults	
<ul style="list-style-type: none"> Minor 60% Major 30% Fatal 10% 		slip through the implementation phase	

Figure 3.4. Example abstraction sheet

Interviews should be recorded for future use and as a feedback mechanism to the interviewee. Such an interview report contains the results of the interview and can include an abstraction sheet of the issues raised in the interview.

Activity 8-3	<i>Define questions and hypotheses</i>
-------------------------------	---

- Analyse interview reports
- Specify GQM questions
- Specify hypotheses

With respect to the measurement goals, questions must be defined to support data interpretation. As the goals are defined on an abstract level, the questions are refinements of the goal that lead to a more operational

level suitable for interpretation. By answering the questions, it should be possible to conclude whether the goal is reached.

For each question, *expected* answers are formulated as 'hypotheses'. Formulating hypotheses encourages the project team to think about the current situation and therefore stimulates a better understanding of the process and/or product. Furthermore, during data interpretation, these hypotheses on measurement results are compared to the actual measurement results. The purpose of this is not to evaluate the possible correctness of the hypotheses, but rather to encourage the project team to identify and analyse the underlying reasons that caused results to conform to or deviate from their expectations.

To ensure that the right questions and hypotheses have been recorded and correctly formulated, a review must be made. The questions are the basic translation from goals to metrics. When the actual data is collected and presented to the project team, it should help in answering the questions of the project team. In this way, the questions take a central role, not only during definition but also during interpretation. Therefore it is important to make sure that the questions are correct. The questions were also reformulated from the project team's input during the interviews. It is possible that mistakes were made during transcription, or that the PROFES team misinterpreted the questions.

The hypotheses must be reviewed as well, as the hypotheses are used together with the questions to define the metrics to be established for data collection.

**Activity
8-4*****Define and check metrics***

- **Analyse GQM questions**
- **Specify the metrics required to answer those questions**
- **Compare metrics with process and product models**
- **Incorporate necessary changes**

Once the goals are refined into a list of questions, the metrics must be defined to provide all the quantitative information for answering the questions in a satisfactory manner. Therefore, metrics are a refinement of questions into a quantitative level of abstraction that identifies process and/or product measurements. Once all these measurements are collec-

ted with respect to the defined metrics, sufficient information should be available to answer the questions completely.

Furthermore, factors that influence the outcome of the metrics must also be defined: factors that directly influence metrics also influence the answers to the questions that the metrics are related to. If these influencing factors are not considered when defining the measurement programme, incorrect conclusions on interpretations of the collected data may be drawn. These influencing factors are also usually defined as metrics.

The defined goals, questions, and metrics must be consistent and complete in relation to the process and product models of the respective project (see Figure 3.5). To ensure this, consistency and completeness checks have to be performed throughout the entire definition phase. If definitions appear to be missing, incomplete, or inconsistent, either the definitions have to be adjusted to comply with the process/product models, or the process/product models have to be adjusted to comply with the goal, question, and metrics definitions.

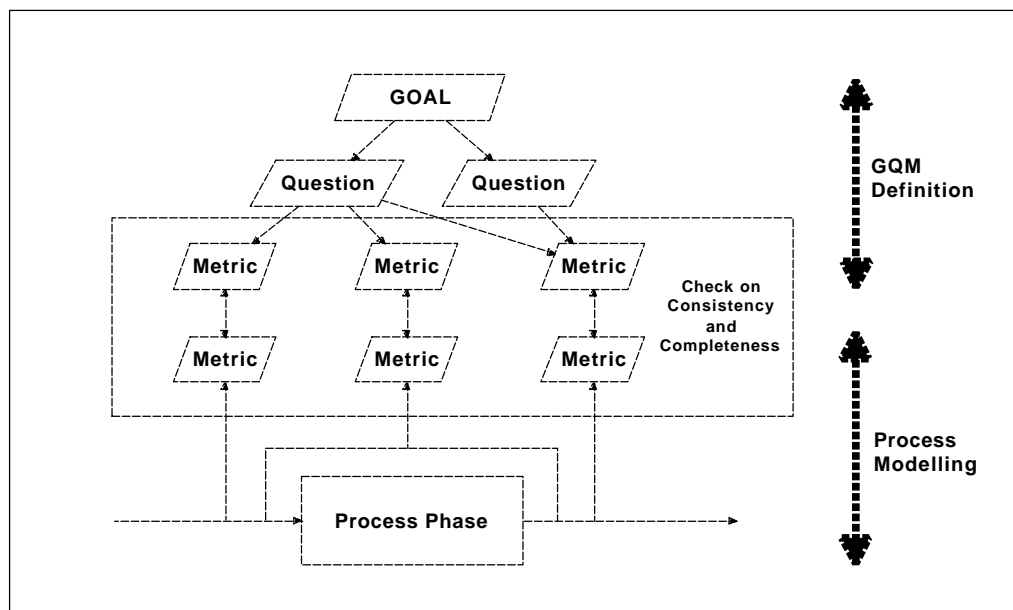


Figure 3.5. Checking for consistency and completeness

**Activity
8-5*****Produce GQM plan and measurement plan***

- **Assemble GQM goal, questions, hypotheses, and metrics**
- **Document them in a GQM plan**
- **Operationalize each metric**
- **Document in a measurement plan**

A GQM plan is a document that contains the goals, questions, metrics, and hypotheses for the measurement programme, as defined in the previous steps. The GQM plan serves as a guideline for data interpretation, and provides a basis for the measurement plan and analysis plan.

The GQM plan describes the refinement from measurement goals into questions and from questions into metrics. As some of these metrics may be calculated (indirect metrics) using other metrics, all direct measurements that will also be collected, are listed in the GQM plan.

A measurement plan is a document for each identified measurement in the GQM plan that describes who should collect it, when it must be collected, and how it must be collected. Furthermore, all automated data collection tools and manual data collection forms are defined in the measurement plan.

The measurement plan describes the following aspects for each indirect measurement identified in the GQM plan:

- It provides a formal description of the direct measurement
- It provides a textual description of the direct measurement
- It defines all possible outcomes (values) of the direct measurement
- It identifies the person that collects the direct measurement, i.e. a programmer, engineer, project manager, tester etc.
- It defines at which specific moment in time this person must collect the direct measurement
- It defines by which medium (tool or form) this person must collect the direct measurement.

As the GQM plan and measurement plan represent the formal definition of the measurement programme and describe all related data collection procedures, they must be reviewed and approved by all project members before data collection can actually begin.

The review session should focus on:

- Do the project members agree upon the defined goals, questions, and metrics?
- Can the project members identify any missing or unnecessary definitions?
- Do the project members agree with the proposed data collection procedures and tools?

Average Duration and Effort

The duration for this step is about one month with an effort of about 8 person weeks.

Tools and Templates

- GQMAAspect, to be used for the definition of a measurement programme
- MetriFlame, to be used to perform data analysis of the measurement data, and to provide feedback to the software development team.
- SW engineering tools, such as compilers or static analysers, since these tools can provide plenty of useful data and can therefore be a practical tool for collecting measurements.
- Spreadsheets, to be used to perform some basic data analysis in trial situations.
- PROFES provides templates for a GQM plan and for a measurement plan in the appendix of this manual.

Work Products

Input work products	Output work products
<ul style="list-style-type: none"> • Prescriptive process model (including selected process changes) • Product quality needs and target profile (result from Step 2) • Current status of product quality (result from Step 3) • Product improvement goals (result from Step 5) • Process assessment reports and profiles (result from Step 4) • PPD models (result from Step 6) 	<ul style="list-style-type: none"> • GQM abstraction sheets • GQM plan • Measurement plan

Resource Allocation

Roles, responsibilities and requested skills

Expert roles

PROFES expert

The PROFES experts are responsible for managing and performing all GQM activities. The PROFES team involves the project development team as much as possible, but however makes sure that the effort spent by the project team remains acceptable.

Support roles

Project team

The project team provides all necessary information for proper GQM planning. This means participation in goal definition, interviews, hypothesis formulation, and document reviews. However, the project team does not carry out operational GQM tasks.

Expected effort/role

PROFES experts

Depending on the experience of the PROFES team, the effort needed for GQM definition is 4 weeks for an experienced PROFES team, and 3 months for an inexperienced PROFES team. In case the project team is less than 10 engineers, this effort can be lower.

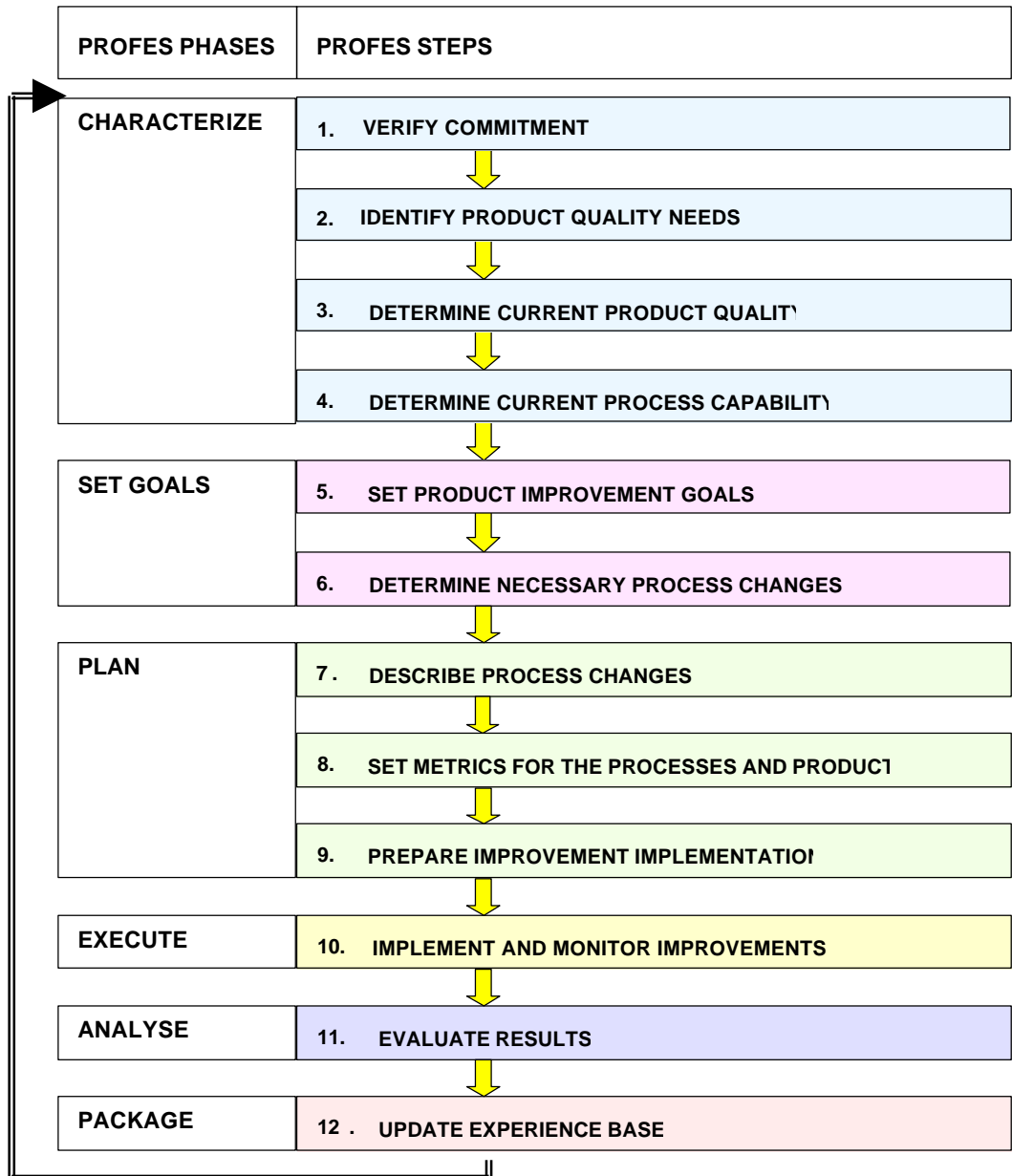
Project team

The project team spends about one day per team member on GQM definition.

Methods and Techniques

Goal/Question/Metrics (GQM) method:

- For short introduction into the principles see Basili, V., Caldiera G. & Rombach, H. 1994. Goal Question Metric Paradigm. In Marciniak, J.J, (ed.), Encyclopaedia of Software Engineering. John Wiley & Sons, Vol. 1, pp. 528-532.
- For practical guidelines and examples see Solingen R., Berghout E. 1999. The goal/ question/ metric method, a practical method for quality improvement of software development. McGraw-Hill ISBN 007-709553-7.



STEP**9****Prepare improvement implementation****PROFES****9-1 Plan process improvement progress meetings****9-2 Make time planning and resource allocation****9-3 Kick-off process changes****Purpose**

The planning of improvement action ensures that selected process improvements will actually be implemented. All prerequisites for the successful introduction of process changes are arranged in this step, resulting in a detailed action plan for the process improvements.

Step goals

- Plan process changes and allocate sufficient resources to implement them
- Plan process improvement progress meetings, which are held by a committee responsible for tracking the progress of the process changes according to plan

Activities

Improvement implementation needs to be well prepared, since it is an important part of an improvement programme. Preparation means fulfilling all prerequisites for successful implementation, and in the context of PROFES consists of:

- Planning meetings to monitor improvement progress
- Making a time planning and resource allocation
- Kicking off the process changes at a meeting

**Activity
9-1*****Plan process improvement progress meetings***

- **Appoint chairman for the improvement progress meetings**
- **Select key people to attend the improvement progress meetings**
- **Plan meetings**

In order to ensure that selected process changes will be implemented and applied, it is necessary to hold regular improvement progress meetings. The process improvement progress board, an appointed committee, will hold these meetings. Its main task will be to track improvement progress, and define corrective action if necessary. We recommend that this board will include representatives of different organizational functions, including project personnel, quality assurance representatives, and senior management representatives. Top management should be adequately represented in the progress meetings, as their commitment will be a key factor for the success of process improvement. If possible, this board will meet every two to four weeks, to discuss progress and results. We recommend that the chairman of these meetings should be a manager, preferably from outside the development group, for example a quality assurance manager, SEPG manager, or PROFES team manager.

**Activity
9-2*****Make time planning and resource allocation***

- **Specify the necessary improvement steps and sub-steps**
- **Co-ordinate the improvement steps with project work**
- **Make an implementation plan for process improvement**

An important part of implementing process changes is proper planning of the improvement steps. Process improvement is much more than simply applying a new procedure. Promotion, training, assistance, piloting process changes, guidance, partial implementation etc. are important steps in making process changes work.

Depending on the size, motivation, schedules etc. of the process changes, the level of detail in time planning can vary. For a large department with ingrained resistance to change, the action plan will need to be more thorough and phased than for a small group of motivated engineers.

However, sufficient resources should be allocated to the different tasks and proper planning of the steps should be carried out.

The improvement plan is the major reference to track status and progress at the improvement progress meetings. Therefore the actual steps should be clearly defined in a traceable manner. The plan should be updated in case of changes, which is likely, and process improvement progress meetings should be included in the planning.

As all GQM planning activities are already finalized, the measurement activities should be included in the planning as well. Feedback sessions can be scheduled to take into account the relevant project milestones. Schedule of feedback session should be included in the improvement planning. Through inclusion of the GQM measurement tasks in the improvement schedule, full integration of measurement and process improvement will be established.

**Activity
9-3*****Kick-off process changes***

- **Invite people to the kick-off meeting**
- **Hold meeting in which the process changes are started**
- **Provide on-line process support**

After appropriate planning, definition of the altered procedures, and assignment of responsibilities for the changes, the actual implementation can begin. We recommend organizing a so-called opening or kick-off meeting' in which implementation of the changes is begun. At such a session, an overview of process improvements and planning may be presented. Such a session again helps in increasing the motivation to carry out process changes. We therefore recommend that as many software engineers involved in the process improvements as possible are present at this kick-off meeting. Naturally, the process improvement progress board should also be present.

On-line process support should also be set up. In case people have questions regarding implementation of the changed process, they can consult this process support service. Such support is very important, especially in the early stages of a new process. Implementation will differ in different situations, but in many cases, appointing a process champion' will be sufficient. In other cases it might be necessary to install an on-line help desk. The way in which this on-line process support is implemented should be decided in accordance with the situation, changes, organization, number of people involved, etc.

Average Duration and Effort

2 – 8 weeks duration, about 15 days of effort

Tools and Templates

- Planning tools and automated procedure/workflow tools
- PROFES improvement plan template, which is included in the appendix of this manual

Work Products

Input work products	Output work products
<ul style="list-style-type: none"> • Development project plan • Preliminary improvement plan (result from Step 4) • Selected list of process changes (result from Step 6) • Prescriptive process model (result from Step 7) • GQM deliverables (result from Step 8) 	<ul style="list-style-type: none"> • Process improvement action plan • On-line process support

Resource Allocation

Roles, responsibilities and requested skills

Managerial roles

PROFES Team Manager

The PROFES team manager is responsible for ensuring that the deliverables of this step are correct and serve their purpose. He/she is also likely to be chairman of the process improvement progress meetings, which involves selecting candidates for this board and briefing the other board members on their roles and tasks. The PROFES team manager should also set-up a schedule for progress meetings, which will also be included in the process improvement schedule.

Project Manager

Project managers are involved in discussing how and when process changes should in practice be implemented in the project they are responsible for. They should review the deliverables of this step, and we recommend that they take part in the improvement progress meetings.

Expert roles

PROFES expert

The PROFES expert is responsible for the development of the main deliverables of this step, the process improvement action plan, and new or revised procedures.

Development of these deliverables is not an individual off-line activity, but should be carried out in full co-operation with development projects and managers. The PROFES expert assigned to this task should therefore be socially skilful.

Support roles

Project team

Project team members are consulted on implementing process changes in the new or revised procedures. Project team members should also participate in the opening session. We recommend that one or two project engineers take part in all main decisions for the process improvement programme.

Process support engineer

Process support engineers support the application of the new processes. These engineers will monitor correct implementation of the process, and will provide support to the project team members in case of question or problems.

Expected effort/role

PROFES expert

The PROFES expert produces most of the deliverables in 40 hours of effort, depending on the number of process changes to be processed in the procedures.

PROFES Team Manager

The PROFES team manager is mainly involved in the meetings of the process improvement progress board. The PROFES team manager also needs to review the deliverables of this

step, amounting to about 12 hours of effort, depending on the amount of process changes included in the procedures.

Project Manager

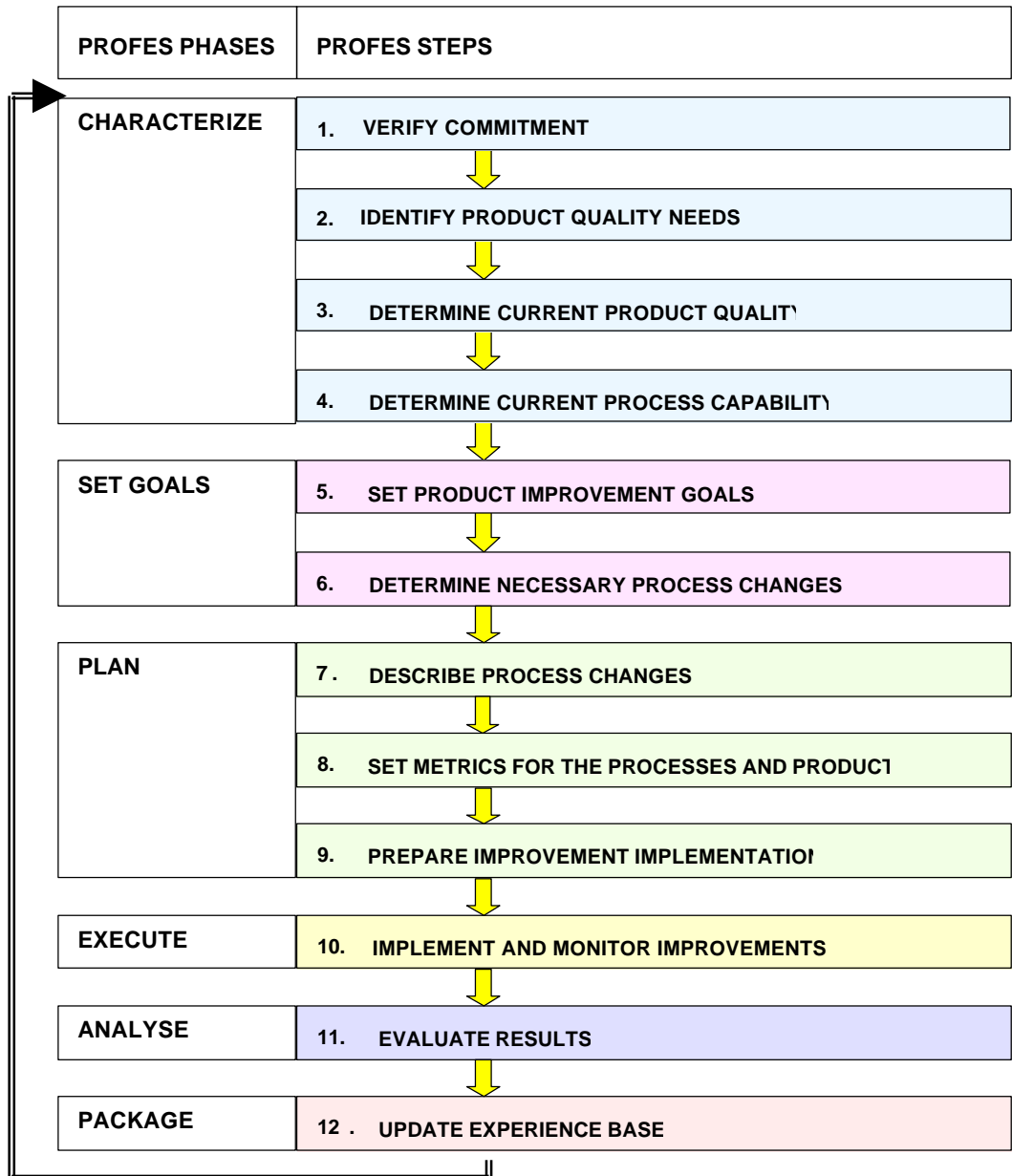
The project manager is mainly involved in the meetings of the process improvement progress board, and reviews documentation, which needs about 8 hours of effort.

Project team

The project team attends the kick-off meeting, which takes about two hours. Some project team members are also involved in the definition or revision of procedures, taking 4-8 hours for each engineer involved.

Methods and Techniques

Process modelling, procedure description techniques, project planning methods and techniques.



STEP**10****Implement and monitor improvements in the development project****PROFES****10-1 Implement process changes****10-2 Collect measurement data****10-3 Prepare and select the measurement data****10-4 Hold GQM feedback sessions****Purpose**

During development project implementation, process changes are made and the data defined in the GQM plan is collected according to the measurement plan. When preparing the data, collected measurements are studied, organized and pre-analysed for the GQM feedback session. The development project members will interpret collected data in GQM feedback sessions.

If the continuous assessment technique is implemented as described in Chapter 4, it is possible to collect data on process maturity and capability during this step.

Step Goals

- Implement selected process changes according to process improvement plan
- Collect the data and prepare the measurement results for each feedback session
- Arrange feedback sessions

Activities

- Implement process changes
- Collect measurement data
- Prepare and select the measurement data
- Hold GQM feedback sessions

**Activity
10-1*****Implement process changes***

First, communicate process changes to the development project in kick-off meeting, as described in the previous step.

Implement process changes according to the process improvement plan, as described in the prescriptive process model. Process changes are usually tested in a single development project before being adopted in all projects.

Process improvement should take place step-by-step, and gradually introduce new changes to the process. The process improvement plan should therefore have a stepwise approach where each process change can be monitored and modified if necessary, before other process changes are made.

All resources needed to implement the change must be available, including training, documentation, tools etc. Organizations may have process support personnel who will help to implement the changes. Training is especially important if there are new methods or tools that are introduced to the development project. Documentation, including guidelines etc. should be used to support the process changes, however it should be remembered that guidelines in themselves are not sufficient without training.

Plan the implementation of process changes carefully and co-ordinate them with the development project plan. The process improvement plan should be made flexible, should there be delays or resource changes in the development project.

It is important to motivate personnel; various motivation mechanisms should be evaluated and selected. Process changes should be planned so that they also improve the practical needs of the development project. In this case, motivation of people is easier, as they can see the results of process changes in their everyday work.

Deployment of process changes to the entire organization requires accurate planning. Usually organizations have software process guidelines for development projects; these should be updated according to the process changes and used as a means to implement process changes. Communication throughout the organization is required in order to deploy process changes. Development projects may have different needs, and therefore process changes have a different impact on different projects.

**Activity
10-2*****Collect measurement data***

- **Collect the measurement data according to the measurement plan**
- **Validate measurement data**

Collect the measurement data according to the measurement plan defined in Step 8.

Allocate the necessary resources for measurement collection in the development project. If possible, collect the metrics automatically or derive them from existing metrics. The PROFES team is responsible for implementing measurement collection. Measurements can be collected from existing databases, reports etc.

Typical collection points include project milestones, end of months etc. It is important that measurement collection does not become a burden for the development project. The following rules of thumb can be used:

- Communicate measurement objectives to the development project personnel asked to collect data, so that they are motivated to do this extra work.
- Collect measurements automatically, using existing databases, tools and reports where possible.
- Collect measurements as seldom as possible.
- Collect measurements that prove to be helpful for the development project's everyday work. This is important for motivation.

When measurements are collected, there may still be some open issues related to measurements. Measurement collecting at an early stage should be emphasized. By solving possible conflicts and misunderstandings in good time, measurement collection will be easier later on.

Open issues may exist especially in subjective measurements such as those using values like 'excellent, good, normal, poor'. People may have different views on how the value should be determined. As an example, if we measure document reusability and define it as 'good', are we referring to any possible line changes, or the time needed to modify reusability? Open issues exist in all measurement programmes, no matter how well they are defined.

Somebody from the development project should be made responsible for measurement procedures, validating them, and discussing open issues with project members and the PROFES team. Typically, project quality managers are also responsible for measurement.

**Activity
10-3*****Prepare and select the measurement data***

- **Prepare the collected data for the feedback session**
- **Select measurements for the feedback session**

The PROFES team prepares the collected data for the feedback session. Preparation is necessary, as there are usually several measurements defined with various views on them, and not all the data can be dealt with at the feedback session. Therefore it is necessary to check which measurements are most useful and focus on them.

Preparation begins by gathering the measurement data available. The measurement data is then processed with statistical or spreadsheet programs etc. to create different views and combinations of measurements. It is possible to initially analyse measurement data at this point, but a more thorough evaluation of measurements is included in Step 11. At this stage, measurements are analysed for the feedback session.

Measurement data describing product quality and process improvement is analysed in preparation for the feedback session. Chapter 4 contains examples of analysis techniques. Data confidentiality, variations, unexpected findings etc. should also be considered in preparation for a feedback session. It is important to notice that preparation does not include interpretation, which is carried out by the development project members at the feedback session.

After the measurements have been prepared, select measurements to be presented during the following feedback session. Incomplete or unreliable data should not be presented during the feedback session.

After the measurements to be presented are selected, create slide presentations for feedback session. If automatic tools are used to present measurements, it is possible to present different kind of views rapidly in feedback session using portable computer etc.

**Activity
10-4*****Hold GQM feedback sessions***

- **Arrange the feedback session**
- **Interpret the measurement results**
- **Document the results**
- **Implement possible changes**

Feedback sessions are organized meetings involving members of the development project and the PROFES team. Feedback sessions provide an essential mechanism for supporting analysis and interpretation of the measurement results.

The main objective of feedback sessions is to discuss the findings and results of the collected measurements, and to derive interpretations by the project team from the data collected so far. It is important that all development project members are present at the feedback sessions.

We recommend performing feedback sessions on a regular basis, at least every other month. In some cases, three to four feedback sessions may be necessary during a six-month period.

Some rules of thumb for feedback sessions:

- Arrange feedback sessions at least every other month, otherwise the project will lose interest in measurements.
- Do not organize a feedback session if there is not enough data available. Incomplete measurement presentations will not encourage development projects to participate in feedback sessions in the future.

Development project members interpret measurement results, and it is important to hear their opinions on collected measurements. All comments, open issues, actions etc. should be recorded in the session minutes.

It is important to discuss the measurements themselves at the feedback sessions. For example, if some measurements are hard to collect or incomplete, the measurement plan can be changed, unnecessary measurements removed, or new ones added. Feedback sessions can also

be used to evaluate PPDs using data from product and process measurements.

Integration of assessment and measurement activities makes it possible to expand GQM feedback sessions to integrate feedback sessions that also discuss process changes and their impact. New process changes may be changed or even rejected. Further information on the integration of assessment and measurement can be found in Chapter 4.

Document all changes in appropriate documents, such as the GQM plan, measurement plan, prescriptive process model, process improvement plan etc.

Continuous assessment can also be used. According to the principle of continuous assessment, measurements collected during development project implementation are used to evaluate process. Further information on this technique can be found in Chapter 4.

Average Duration and Effort

Duration depends on the development project. Measurement collection typically requires at least a few months.

Effort is required from the development project to implement process changes and collect measurements. Effort for process changes varies depending on the changes made, but it should be noted that some training and familiarization with new practices is necessary. Effort for measurement collection is usually a few hours for each project member, depending on the number of "collection points." Each feedback session requires two to three hours'effort from each development project member.

PROFES team effort is necessary for creating a measurement infrastructure and analysing the results. During the PROFES project, the effort required was as follows, expressed in person days per feedback session.

Activity	PROFES team	Development project
Data collection	1 person-day	3 person-days
Data analysis and interpretation	5 person-days	2 person-days

The figures above are only estimates. Actual effort may vary according to experience, the infrastructure available, and the number of measurements defined.

Tools and Templates

- Measurement collection tools such as MetriFlame, statement counting tools, fault reporting systems etc. Information about PROFES tools including MetriFlame is available in Appendix 5 of this user manual.
- Data collection forms or panels to insert data into databases
- Commercial statistical software packages or other analysis programs
- Visualization tools (e.g. Excel, PowerPoint, MetriFlame)
- Assessment Tools, e.g. trend analysis tool, presented in Appendix 5.
- Template for feedback session report
- Overhead page templates for visualization

Work Products

Input work products	Output work products
<ul style="list-style-type: none"> • Prescriptive process model • GQM plan • Measurement plan • Process improvement plan • Development project plan 	<ul style="list-style-type: none"> • Measurement data • Feedback session report(s) with visualized measurement data • Description of corrective actions taken • Prescriptive process model applied in practice

Resource Allocation

Roles, responsibilities and requested skills

Managerial roles:

Project Manager

Responsible for implementing and reporting on the project

Project managers interpret the collected measurements and are essential for successful process improvement. Depending on

the development project and defined measurements, their role in measurement collection may also be vital.

Project quality manager

Responsible for project quality.

Collects and/or validates the data. Solves open issues.

Expert roles:

GQM expert/PROFES team

Act as GQM method facilitator

Creates the means for collecting measurements, such as templates and database connections. Selects and analyses the collected measurements for feedback sessions

Project members

Act as data collectors and interpreters.

Collect data, interpret collected data, and draw conclusions.

Support roles:

Process support (SQA) person

Implements the process changes.

(Optional role) Statisticians

Act as Statistical experts in large projects.

Expected effort/role

Please note that the effort for process changes is not included here, as it depends completely of the process changes themselves.

Managerial roles:

Project manager

Total effort is approximately 3-4 person days. Project manager participates in data collection and interprets the results.

Project quality manager

Solves any open issues arising from the development project and implementation of measurement collection and process changes. Excluding the process change effort, this requires a few days, perhaps 3-4 person-days.

Expert roles:

GQM expert/PROFES team

The GQM expert prepares the data for feedback sessions, provides the means for data collection, etc. His effort for each feedback session is about 5-6 person-days.

Project members

Their effort is needed for data collection and interpretation of the results. This may typically take some 2-3 days for each feedback session.

Support roles:

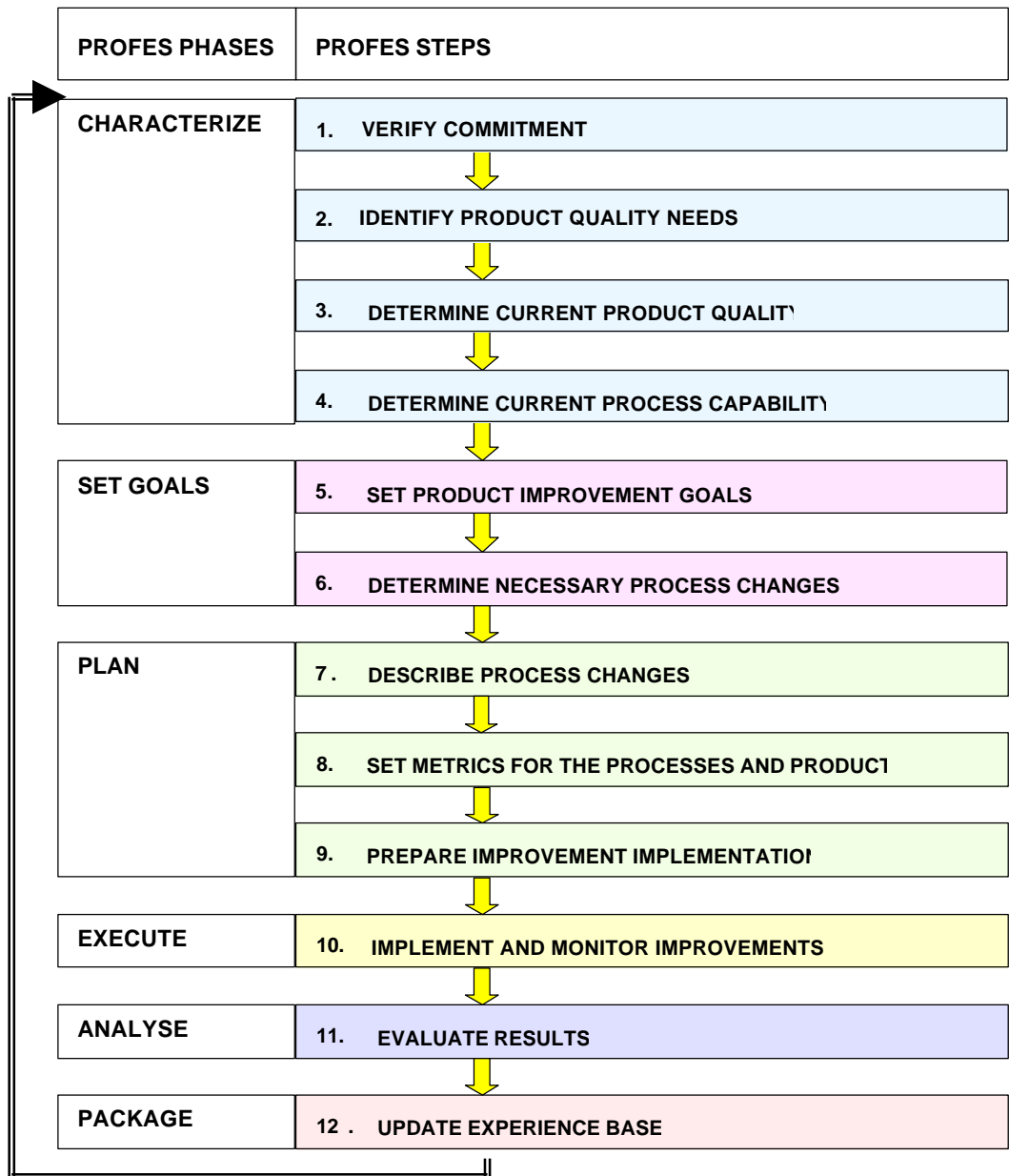
(Optional) Statisticians

Their effort is only necessary in very large projects. Depending on the analysis methods used, the effort is a few days per analysis.

Methods and techniques

- Goal/Question/Metrics (GQM) method:
 - For short introduction into the principles see Basili, V., Caldiera G. & Rombach, H. 1994. Goal Question Metric Paradigm. In Marciniak, J.J, (ed.), Encyclopaedia of Software Engineering. John Wiley & Sons, Vol. 1, pp. 528-532.
 - For practical guidelines and examples see Solingen R., Berghout E. 1999. The goal/ question/ metric method, a practical method for quality improvement of software development. McGraw-Hill ISBN 007-709553-7.
- Automated or semi-automated measurement collection. See for example: Parviainen P., Järvinen J., and Sandelin T. 1997. Practical Experiences of Tool Support in a GQM-based Measurement Programme. Software Quality Journal, Volume 6, No. 4, December 1997, pp. 238 - 294.
- Statistical analyses (e.g. correlation analysis, regression analysis, trend analysis, P-charts, pareto diagrams etc. See also literature, e.g. Burr A, Owen M. 1996. Statistical Methods for Software Quality.
- Data mining techniques/machine learning. Several books can be found on this subject, for example: Adriaans P., Zantinge D. 1996. Data Mining.; Cios K., Pedrycz W., Swiniarski R. 1998. Data Mining Methods for Knowledge Discovery. Kluwer, ISBN 0-7923-8252-8.; Mitchell T. 1997 Machine Learning. McGraw Hill.

- Personnel motivation techniques
- Effective meeting and presentation techniques
- Assessment methods (e.g. BOOTSTRAP, CMM), Kuvaja P., Similä J., Krzanik L., Bicego A., Saukkonen S., Koch G. 1994. Software Process Assessment and Improvement: The BOOTSTRAP Approach. Blackwell Publishers, ISBN 0-631-19663-3; Humphrey, W. 1989. Managing the Software Process. SEI Series in Software Engineering. ISBN 0-201-18095-2, Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.



STEP

11

Evaluate Results

PROFES

11-1 Evaluate the measurement results**11-2 Support, modify, or reject used PPD models****Purpose**

The purpose is to evaluate the collected product and process-related data in order to find out whether the PPD model or models used were appropriate in the context of the current software development project. Based on this evaluation, the PPD model or models used are supported, modified, or rejected. The purpose is also to gather and analyse data and lessons learned, in order to improve management of product quality-based process improvement projects in the future.

Step Goals

- Evaluate the effect of the improvement programme on the final product quality
- Evaluate changes made to the software engineering process, methods, and tools
- Gather and evaluate lessons learnt
- Evaluate used PPD models using product and process measurement results and lessons learned. Based on this evaluation, PPD models are supported, modified, or rejected.

Activities

During project implementation in Step 10, the measurement data was regularly analysed according to the GQM measurement plan. The purpose of this analysis was to monitor the improvement actions and their effects, and to take corrective action if necessary. Step 11 takes place when the product development project is completed, and all measurement and lessons learned data is available. The final, exhaustive measurement data analysis session is held using this data. Step 11 consists of the two following activities:

- Evaluate the measurement results
- Support, modify, or reject used PPD models

**Activity
11-1*****Evaluate the Measurement Results***

- **Gather all measurements related to the processes, product quality, methods, tools, and lessons learned**
- **Prepare measurement data for analysis session**
- **Evaluate measurement results together with project personnel**

When the software development project is completed, there should be enough measurement data to conduct a thorough post-mortem analysis of its experiments and achievements. However, before the results can be evaluated, check that the necessary measurement data is available and prepare it for analysis. Although data presentations already prepared for previous analysis sessions may be reused, new presentations in different formats such as graphs, tables, or charts will most likely be needed. This is partly due to the fact that complete product-related quality data might only be available after a certain time period has passed since the end of the project.

When all necessary data is available and prepared for evaluation, the analysis session can begin. Evaluate the data collected during and after project implementation, in order to find out whether the project has reached the product improvement goals set at the beginning of the project. Evaluate the measurement data according to the framework developed in Step 8 (Set Metrics for the Processes and Product). The abstraction sheets and the GQM plan (i.e. input from Step 8) describe the anticipated dependence between the product quality measurement data collected, and process performance. Always remember to evaluate the measurement results jointly with the project personnel. Naturally, achievement of product quality goals can be assessed without the project personnel, but then experience, lessons learned, and other important and substantial information necessary for improving the processes, methods and tools used would be ignored.

Document the results and conclusions from the analysis session in the final report, to be used for packaging the improvement project results in Step 12 (Update Experience Base).

**Activity
11-2*****Support, Modify, or Reject Used PPD Models***

- **Evaluate PPD models used**
- **Support, modify, or reject PPD models used**

Once the project has ended, it is time to compare the improvement goals with the actual improvement results. To review and compare the selected or developed PPD models with the attained results is the other main activity of Step 11. This is to evaluate whether the product improvement goals have been achieved by the changes made to the software engineering process, methods, and tools.

Evaluate the measurement results to determine whether or not the measurement results and experiments can substantiate the assumed dependence. Use the dependence descriptions in the form of PPD models, and evaluate their validity in the context of the product development project. Use the results and conclusions of the previous joint analysis session, and decide whether each PPD model used can be supported, or should be modified, or is to be rejected in the context of this project. This evaluation can either be part of the joint final analysis session, or it can be carried out at a separate meeting later.

Modify the PPD model descriptions according to the evaluation results. Add the information gained concerning PPD model usability, achieved results and other experience, in order to enable existing PPD models to be more easily and reliably reused. It may well be that the PPD model used can be fully supported in the software development project. For future use, also update this information in the PPD model description. Normally, more than one PPD model is used to define the process changes necessary, and one of them may have to be rejected, based on knowledge gained during or after the project. All decisions made regarding the used PPD model should be justified, even when rejecting the PPD model.

Average Step Duration and Effort

The ideal step duration is one to two weeks, but should not exceed one month. The effort required is closely connected to the duration of the measurement programme, and the number of personnel participating in the analysis session. The total effort should be less than two weeks.

Tools and Templates

The company can choose different tools for data gathering and analysis purposes. Examples of such tools are:

MS Excel, see B. J. Dretzke 1998. *Statistics With Microsoft Excel*. Prentice Hall College Div; ISBN: 0139565337; P. Carey & K. N. Berk 1997. *Data Analysis with Microsoft Excel*. International Thomson Publishing; ISBN: 0534529291

Data visualisation tools,

- Zinnote, see Zinnote Data Integration and Reporting Toolkit, Getting started. 1998 Positive Support Review Inc. <http://www.zinnote.com/Download/GettingStarted.pdf>; Zinnote Tutorial. <http://www.zinnote.com/TutorialIndex.htm>
- or SPSS, see M. J. Norusis. *SPSS 8.0 guide to data analysis*. Upper Saddle River, NJ: Prentice-Hall, 1998. ISBN: 0-13-687484-3 ;SPSS Bookstore. <http://www.spss.nl/store/bookstore.htm>

Data analysis tools,

- SAS, see K. A. Hardy & F. C. Dilorio 1996. *Quick Start to Data Analysis With Sas*. Duxbury Pr; ISBN: 0534237606, R. P. Cody, R. Cody, J. 1997. *Smith Applied Statistics and the Sas Programming Language*. Prentice Hall College Div; ISBN: 0137436424,
- CART, see *CART homepage (Salford Systems)*. <http://www.salford-systems.com/index.html>,
- Statistica, see *Quick Statistica for Windows*. Paperback Windows 98 edition (January 1, 1999). Statsoft Inc; ISBN: 1884233120; Statistica Home Page (Statsoft). <http://www.statsoftinc.com/index.html>
- or JMP *Jmp Introductory Guide, Version 3.1*. Paperback 3 volume set edition (July 1996). Sas Inst; ISBN: 1555446809

During the PROFES project, MetriFlame was used to gather data from different sources according to the GQM measurement plan, and to prepare data for analysis. More information on Metri Flame can be found in Chapter 6 and Appendix 5 and from references.

Update the PPD models when modification to PPDs is necessary. An example of a PPD model template can be found in Chapter 7 and Appendix 5.

Work Products

Input work products	Output work products
<ul style="list-style-type: none"> • PPD models • Prescriptive process model • Abstraction sheets • GQM plan • GQM measurement plan • Measurement data • Feedback session reports 	<ul style="list-style-type: none"> • Preliminary experience packages • Evaluated PPD models

Resource Allocation

Roles and responsibilities

Managerial roles

Project managers

- § Control the evaluation activities
- § Participate in measurement results evaluation

Quality assurance managers

- § Gather the measurement data and prepare it for analysis
- § Update the PPD models
- § Draw up the final report

Expert roles

Project members

- § Participate in measurement results evaluation
- § Review the final report and updated PPD models

Support roles

Evaluation support

- § Assist evaluation activities, e.g. documentation, organizing meetings, etc.

Expected effort/role*Managerial roles*

1 to 2 weeks

Expert roles

Less than 1 week

Support roles

Less than 2 days

Methods and Techniques

Generally, no formal evaluation methods and techniques are necessary, as evaluation is the result of analysis and discussion meetings. Data analysis techniques such as statistical analysis can be used to assist the evaluation of measurement results. More information about statistical data analysis techniques can be found from many www-sites, for example:

Extensive descriptions of statistical data analysis methods are located at:

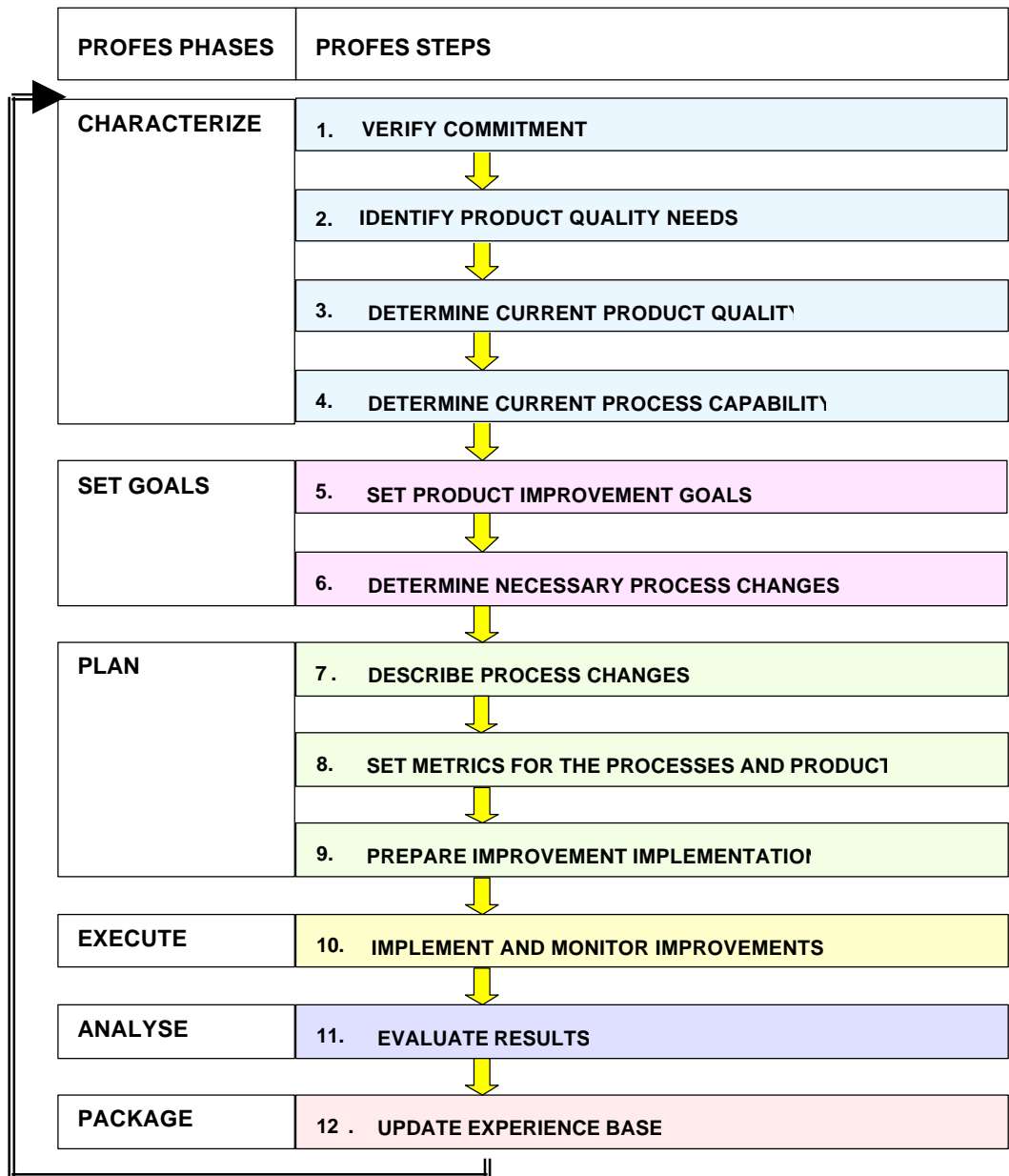
- <http://ubmail.ubalt.edu/~harsham/stat-data/opre330.htm>

Basic information on statistical methods such as correlation, mean, deviation etc. at:

- <http://www.psychstat.smsu.edu/sbk00.htm>
- <http://curriculum.qed.qld.gov.au/kla/eda/>
- <http://www.execpc.com/~helberg/statistics.html>
- <http://www.math.montana.edu/~umsfjban/Courses/Stat438/Text/GDA.html>
(also includes graphical presentation information, scatterplot, dot, charts, etc.):

Links of more general statistical resources on the web, with a section related to teaching statistics:

- <http://www.execpc.com/~helberg/statistics.html>



STEP**12****Update Experience Base****PROFES****12-1 Package information****12-2 Store relevant information in the Experience base****Purpose**

The purpose of this step is to save all experience gained from the project. This also includes rejection and modifications, which were made in Step 11 (Evaluate Results). Storage of experience is necessary for later reuse in forthcoming projects, especially when performing Step 6 (Determine necessary process changes) in the next project.

Step Goals

Package and store all information gained during the project in the Experience base for future reuse.

Activities

Software engineering experience can be any kind of data, information, or knowledge that is relevant for future software projects or improvement programmes. To be reusable, software engineering experience should be described explicitly and stored in an accessible repository, here referred to as the Experience base. Typical kinds of experience are prescriptive process models, product-process dependence models, quantitative profiles or prediction models, and informal, narrative "lessons learnt" - statements. This step can be carried out by performing the following two major activities:

- Package Information
- Store relevant information in the Experience base

**Activity
12-1*****Package Information***

- **Identify future reuse needs**
- **Define models in reusable form**
- **Define future reuse context**
- **Package model with context definition**

In order to be reusable, newly gained experience needs to be packaged. Packaging involves abstraction and formalization (or structuring) of information, as well as describing the application domain of the reusable artifact. The form of the reusable artifact should allow for its efficient adaptability in the reuse situation. The domain definition is the basis for identifying and retrieving prospective reuse. Experience packages can be represented in various forms, based on paper or electronic media. Although electronic representation is advantageous with regard to flexibility and retrieval support, paper-based media can still be more appropriate in many cases. The appropriate representation form for a given experience package depends on the kind of experience, its usage and accessibility requirements, modification needs, available infra structure, user preferences, and organizational culture. Examples of representations of software engineering experience are handbooks, presentation slides, Intranet pages, databases, or knowledge-based systems. In most cases, it is useful to combine and integrate several forms of representation. For instance, a standard development process can be documented as a handbook and at the same time can be presented via Intranet pages. The Intranet pages can be linked with a database, so that typical effort distribution in variants of the standard development process can be accessed.

For packaging experience from improvement programmes, the following kind of information is particularly relevant:

- Measurement data and measurement programme documentation
- Interpretation of the measurement data with respect to the GQM goals, as carried out during feedback sessions
- Suggested improvements of
 - § Products
 - § Processes

§ Quality models

- Validated dependence between process and product (PPDs)

**Activity
12-2*****Store Relevant Information in the Experience base***

- **Build infrastructure for the Experience base**
- **Store packaged information in the Experience base**
- **Disseminate packaged information throughout the organisation**

If an Experience base infrastructure is already available, then the relevant information (packaged according to the previous activity) can be stored. Otherwise, the responsible roles and the technical infrastructure have to be set up.

The technical infrastructure of the Experience base to be implemented depends largely on the way lessons learned are to be disseminated within the organization:

- If no tool support is available, lessons learned may be collected in binders ready for retrieval on demand. This requires that new lessons learned are copied and distributed to project managers, who are expected to use them.
- An alternative is to write a report, which describes both a specific object (e.g. the GQM process) and the lessons learned related to the object. It would be better to have this report on-line, as then the update distribution cycle can be shortened.
- The most advanced alternative is tool support for on-line access over a www-based network for visualization of stored knowledge data. Dissemination takes place automatically as new lessons learned become available when entered into the system.

The packaged information is stored using one of the suggested alternatives or existing practices in the organization.

After the packaged information has been stored, the information about these new experience items has to be disseminated throughout the organisation. This can be done for example via organisation-internal newsgroups, the organisation's intranet or by announcements at

department meetings. Furthermore, the information can be included in briefing and training materials, which are used for projects.

Average Duration and Effort

The expected effort depends largely on whether an infrastructure for the Experience base is available or not. If the infrastructure is already in place, the overall effort for this step should be about one to two days.

Tools and Templates

Tool support for the Experience base is not mandatory, but would increase the efficiency of storing, maintaining, and retrieving experience packages and associated information. In general, the following tools and templates could be of use in this step:

- Paper-based collection of material in binders
- Web-based collection of relevant material
- Intranet technology and/or document management systems

More specialized templates and prototype tools for this step are:

- Template for experience packaging
- KONTEXT: A tool that implements an Experience base and assists in searching it, based on similarity searches and decision support methods, for further information, Andreas Birk, Felix Kröschel, "A knowledge management lifecycle for experience packages on software engineering technologies," in Proceedings of the Workshop on Learning Software Organizations, Kaiserslautern, Germany, June 1999
- INTERESTS: A tool for constructing, maintaining, and using the Experience base, based on case information retrieval (for further information: Klaus-Dieter Althoff, Andreas Birk, Christiane Gresse von Wangenheim, and Carsten Tautz. "Case-Based Reasoning for Experimental Software Engineering." In M. Lenz, B. Barsch-Spörl, H.-D. Burkhard, and S. Wess, editors: "Case-Based Reasoning Technology – From Foundations to Applications," pages 235–254. Springer-Verlag, Berlin, 1998.

Work Products

Input work products	Output work products
<ul style="list-style-type: none"> • Evaluated PPD models • Experience base • Process models • GQM plan • Feedback session reports 	<ul style="list-style-type: none"> • Updated Experience base with generalized: <ul style="list-style-type: none"> § PPD models § Process models § GQM Plans

Resource Allocation

Roles and responsibilities

EB Supporter

The EB Supporter records new experience and supports the project teams.

EB Engineer

The EB Engineer packages and analyses existing experience.

EB Manager

The EB Manager provides resources, defines strategic goals, and initiates improvement programs.

EB Librarian

The EB Librarian is responsible for creating and maintaining the Experience base.

Each of these four roles can be assigned to the PROFES expert.

Expected effort/role

EB Supporter

The expected effort for the EB Supporter is less than one day.

EB Engineer

The expected effort for the EB Engineer is less than one day.

EB Manager

The expected effort for the EB Manager is less than one day.

EB Librarian

The expected effort for the EB Librarian is less than one day in the case that Experience base infrastructure is already in place. If not, then the effort depends on the alternatives chosen for storing the information.

Methods and Techniques

The following additional material describes in more detail some of the methods and techniques used in this step:

- "The Experience Factory: How to build and run one" Tutorial by Victor Basili and Frank McGarry, given at the 19th International Conference on Software Engineering (ICSE 19), Boston, USA, May 1997.
- "Knowledge Management of Software Engineering Lessons Learned" Report by Andreas Birk and Carsten Tautz, presented at the 10th International Conference of Software Engineering and Knowledge Engineering (SEKE 98), San Francisco, USA, June 1998.
- "Modelling the Application Domain of Software Engineering Technologies" Report by Andreas Birk, presented at the 12th International Conference on Automated Software Engineering, Incline Village, USA, November 1997.
- "A Knowledge Management Lifecycle for Experience Packages on Software Engineering Technologies" Report by Andreas Birk and Felix Kröschel, presented at the Workshop on Learning Software Organizations, Kaiserslautern, Germany, June 1999.



Chapter 4

HOW TO USE ADVANCED ASSESSMENT AND MEASUREMENT TECHNIQUES IN PROFES

Chapter 4 presents advanced assessment and measurement techniques that can be used in organizations to obtain additional benefits from the PROFES improvement methodology. The following three topics are covered: Integration of software process assessment with goal-oriented measurement, continuous assessment, and advanced data analysis techniques.

To the Reader

The purpose of this chapter is to provide the reader with a detailed description of the advanced assessment and measurement techniques available in addition to the PROFES steps. Together with the PROFES steps, this chapter provides comprehensive guidance for applying the PROFES improvement methodology.

Target audiences for this chapter are organizations already familiar with software process assessment and goal-oriented measurement methods, and who wish to further benefit from the PROFES improvement methodology.

Organizations unfamiliar with software process assessment and goal-oriented measurement should use the PROFES step-wise approach as described in Chapter 3. Once they have more experience with software process assessment and goal-oriented measurement, they can begin to apply the advanced techniques presented here.

Why Use Integrated Software Process Assessment and Measurement?

Software process assessment and goal-oriented measurement require a good understanding of the organization and projects that they are to be applied to. Techniques for collecting, using, and reporting information from organization and projects have common features. The need to integrate software process assessment and goal-oriented measurement is based on this approach. The objectives for an organization applying integrated process assessment and goal-oriented measurement are as follows:

- To complement the views of assessors and Goal/Question/Metric (GQM) experts and make their work more effective
- To make best use of the engineers' time
- To avoid redundant activities that increase improvement costs

The purpose of this chapter is to demonstrate how software process assessment and goal-oriented measurement can be performed in an integrated way. In the PROFES project, assessments were made using the ISO 15504-compliant BOOTSTRAP assessment methodology, and measurements were made according to the GQM method. This section deals with these two methods, although integration can generally also be conducted with other assessment and measurement methods.

Integration Principles

Preliminary product improvement goals assist and focus the forthcoming organizational and project level assessments. These are carried out with the ISO15504 process assessment method and measurement definition according to the GQM method respectively.

The processes to be selected are assessed by applying specific product-process dependency (PPD) models that indicate which processes are most likely to have an influence on the required product quality. By using these PPD models, it is possible to focus solely on the most critical processes. Focused assessment offers a good starting point for a preliminary measurement programme to be run in conjunction with assessment. More information on PPDs can be found in this user manual.

When discussing the integrated use of measurement and software process assessment, two main principles can be defined (see Figure 4.1):

1. By integrating the preparation phase, interviews, and feedback, assessment and measurement activities in the improvement project can be combined.

2. Measurement data collected by the software development project for assessing the process capability can be used both during and after the software development project.

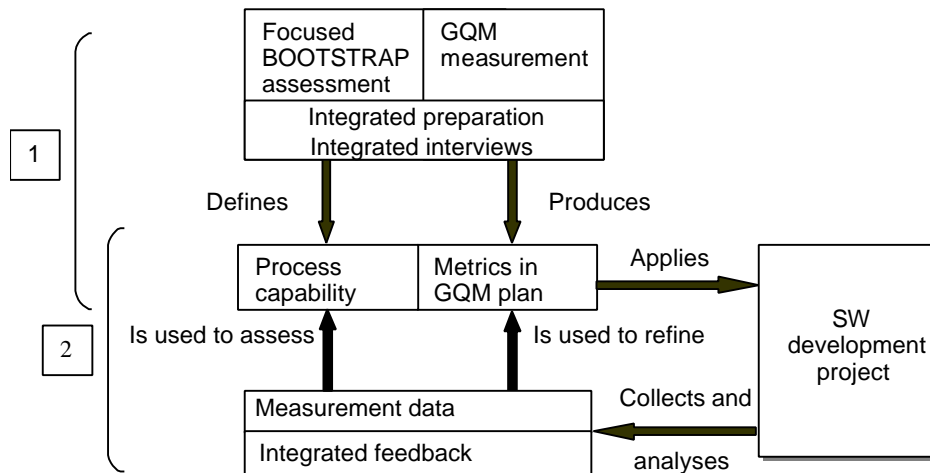


Figure 4.1. Principles of the integrated use of software process assessments and measurements

The integration of activities is described in Chapter 4.1. The use of measurement data for assessing processes is discussed in Chapter 4.2.

Integration and the PROFES Improvement Methodology

Step 4 of the PROFES improvement methodology steps contains software process assessment activities, and Step 8 contains measurement activities. Process changes are implemented during Step 10 and measurement data is collected.

Integration of assessment and measurement activities is closely related to these steps. The purpose of integration is not to replace the step-wise approach of PROFES, but instead to combine and conduct certain activities of software process assessment and measurement earlier than the PROFES step-wise approach would otherwise recommend.

Integration is suitable for organizations and projects familiar with software process assessment and measurement, and does not require any substantial effort to begin these activities.

Integrated Activities

The following software process assessment and measurement activities can be integrated:

- Preparation
- Interviews with software producing unit (SPU) project personnel
- Feedback

In this section, we will describe these activities and explain how they can be integrated. There are other activities that can also be carried out during software process assessment and measurement, but as they cannot be conducted in an integrated manner, they will not be discussed here.

Integrated Preparation

Integrated preparation combines assessment preparation and GQM preparation. GQM preparation includes environment characterization and measurement goal definition, both of which are activities carried out during GQM planning.

Table 4.1 describes the integrated preparation activities with more detail:

Table 4.1. Activities carried out during integrated preparation

Assessment ¹	GQM method
Integrated preparation	
<ul style="list-style-type: none"> ▪ Select sponsor ▪ Define assessment purpose ▪ Identify and characterize the organization and projects ▪ Set assessment scope ▪ Select processes to be assessed 	<ul style="list-style-type: none"> ▪ Identify sponsor and stakeholders ▪ Characterize environment ▪ Set measurement goals
<ul style="list-style-type: none"> ▪ Select assessment team ▪ Select key persons to interview 	<ul style="list-style-type: none"> ▪ Select measurement team ▪ Select key persons to interview
<ul style="list-style-type: none"> ▪ Collect supporting material 	<ul style="list-style-type: none"> ▪ Collect supporting material
<ul style="list-style-type: none"> ▪ Schedule interviews ▪ Define assessment schedule 	<ul style="list-style-type: none"> ▪ Schedule interviews ▪ Define measurement schedule

The sponsor, purpose, improvement goals, measurement goals, and software process assessment scope should all be defined. A sponsor within the organization is required, who is typically a person responsible

¹ Activities in this and other activity tables are based on BOOTSTRAP assessment method. Activities in other ISO15504 compliant assessment methods may vary.

for the SPU and provides the resources and budgeting for product and process improvement.

In general, the purpose of assessment is to aid process improvement, but the specific purpose of the assessment is based on discussions with top management. If measurements are used together with assessment, they should be co-ordinated. Measurements can be used for measuring process improvements or process capability.

A preliminary proposal for SPU-level product improvement goals and measurement goals should be defined. This requires that the organization has been previously characterized and some important issues for analysis have already been identified. This proposal can be used as a basis for defining measurement goals in SPU-level interviews.

Measurement goals can directly reflect business goals or project goals. Measurement goals must be carefully selected, based on selection criteria such as project or organization priority, risk, time available to achieve goal, or potential for understanding, control, or improvement. The better the measurement goals match the improvement goals and assessment scope, the less time and effort is needed to define the measurements.

Assessment scope includes identification of the SPU, projects to assess, and processes to focus on. In the case of a focused assessment, we recommend that processes be selected that will have the greatest impact on the organization's business, product, and process improvement goals. PPDs offer valuable support in helping to select processes. Previous assessment recommendations and ongoing improvement initiatives also provide pointers for identifying prospective processes for assessment.

The PROFES team is responsible for carrying out the complete PROFES cycle. Roles in the team include:

- Lead assessor, who is responsible for the assessment and competent and qualified for this assignment. He or she may either be working in the organization or someone from outside, such as a consultant.
- Assessor, assists the lead assessor and is either an internal or external person.
- GQM expert, responsible for the GQM process.
- Measurement facilitator, an internal person responsible for providing the measurement infrastructure and conducting the measurements.
- PROFES facilitator, an internal person who is responsible for arranging the assessments, interviews, required material, infrastructure, etc.

In most cases, some of these roles could be combined when a complete PROFES software process improvement is to be carried out in a cost-efficient way. For example, a qualified PROFES expert might be able to take on the roles of lead assessor and GQM expert. However, we recommend that there are at least two people in the PROFES team, so that both assessment and measurement aspects are covered as thoroughly as possible. Also for the reliability of the assessment this is usually advisable.

Supporting material is collected to familiarize the PROFES team in advance with the organization and the application domain. It is based on a study of all the relevant documentation, and includes for example:

- Quality manuals
- Process descriptions
- Possible previous assessments
- Possible results of previous measurement programmes
- Project documentation examples
- Product documentation examples

It is important to use existing GQM plans and measurement plans for including additional information on measurements defined during the integrated interviews. Integrated interviews provide a more comprehensive view of measurement than stand-alone GQM interviews, and so existing measurements can be used to complement the GQM and measurement plan.

The PROFES team negotiates an interview schedule with selected interviewees allowing for their availability. A schedule of both assessment and measurement activities is defined together with the sponsor. The resulting plan should contain at least the following information:

- Activity responsibilities
- Interview schedule
- PROFES team roles
- Schedule for possible complementary interviews. A draft GQM plan is created during integrated interviews. Depending on organization/-project's measurement experience, measurement goals and GQM plans may be created based on integrated interviews. Additional interviews may also be required.

- Schedule for review and acceptance of GQM and measurement plans
- Feedback sessions

The selection of interviewees should be included in the plan, and invitations sent to the persons chosen.

Integrated Interviews

Integrated interviews combine assessment interviews held during assessment and GQM interviews held during GQM planning.

The SPU and one or more application projects are assessed. These activities form a major part of the characterization phase in the PROFES cycle. During assessment, interviews are held at both SPU and project level. However, SPU interviews are not obligatory for focused assessment.

GQM interviews are held as part of the assessment interviews, and are usually conducted during the PROFES planning phase. By using integrated interviews, it is possible to combine planning phase activities with characterization phase activities, thus saving time and effort.

Table 4.2 describes the activities of integrated interviews:

Table 4.2. Activities conducted during integrated interviews

Assessment	GQM method
Integrated interviews	
<ul style="list-style-type: none"> ▪ Opening briefing 	<ul style="list-style-type: none"> ▪ Opening briefing
<ul style="list-style-type: none"> ▪ Carry out SPU level interviews 	<ul style="list-style-type: none"> ▪ Review and refine GQM goals if necessary
<ul style="list-style-type: none"> ▪ Carry out project level interviews 	<ul style="list-style-type: none"> ▪ Carry out GQM interviews
<ul style="list-style-type: none"> ▪ Create draft version of assessment report ▪ Review assessment report by the organization ▪ Update and finalize assessment report 	<ul style="list-style-type: none"> ▪ Create draft version of GQM plan ▪ Review GQM plan ▪ Identify additional measurements if necessary ▪ Create measurement plan

Integrated interviews begin with an opening briefing. This is to minimize any potential psychological barriers and explain the plans for the assessment and measurement programme, so that subsequent activities can be carried out effectively and efficiently. All assessment and measurement participants should have a clear understanding of the task’s purpose, schedule, and what is expected from them. At least all those directly contributing to the assessment and measurement activities should be

invited. We suggest that all of those who will benefit and/or will be affected by the assessment/measurement results should be involved.

Interviews are conducted at both SPU and project level in that order. Several viewpoints may be put forward for the SPU interviews, and each viewpoint requires a separate interview. At least one person should be appointed for each viewpoint. Adequate time should also be allocated to each viewpoint as appropriate for the plan. No interview should take more than two hours. Please note that PROFES team roles (assessors and GQM experts) may also be combined thus requiring fewer people.

- The purpose of the assessment viewpoint is to understand important organizational issues such as organizational structure and culture, strengths and weaknesses, business objectives and improvement goals.
- The main purpose of the GQM measurement viewpoint in the SPU interviews is to review the measurement goals and refine them if necessary. The measurement goals are reviewed primarily to ensure their importance for the SPU. They should be clearly related to the important issues in software development for the SPU.

The SPU-level interviews have two main aspects, an assessment aspect and a GQM measurement aspect. The assessment aspect dominates the SPU interviews and the GQM measurement viewpoint dominates the project interviews. The focus of each interview should be defined in advance and explained to the interviewee.

The preliminary measurement goals should be verified and fine-tuned, if necessary. The results can be reviewed at the SPU level before the project level interviews begin. The main purpose is to present the results to SPU interviewees, collect feedback, verify findings, and make corrections, if and when required.

The project level interviews have the same assessment and measurement aspects as the SPU level interviews. However, the measurement aspect dominates at the project level. A draft GQM plan is created according to the measurement goals defined previously. During the interview, the knowledge of people interviewed is recorded to provide information for a GQM plan.

Ideally, the needs of the assessment and GQM planning should be covered in a single interview with each interviewee. Due to limited time, this often means that assessments have to be focused and/or specific details of GQM plan and measurement plan are completed later on.

A preliminary version of the GQM and measurement plans should be developed before project level assessment, if measurements are already

collected already at an early stage in the project. Such measurements might exist in situations where projects are constantly measured with certain sets of metrics. In this case, a second version of the GQM plan is created during the integrated interviews to ensure that the final improvement actions are followed either by measurements or measurements providing data for continuous assessment.

In stand-alone GQM interviews, quality focus, variation factors, and related hypothesis for both the quality focus and the variation factors, are collected for each measurement goal. Due to the broader scope of assessment interviews, it is not possible to collect data in as much detail in integrated interviews. Therefore, there are two possibilities for completing any missing details in the GQM plan:

- a) Additional complementary GQM interviews are organized that validate and add data to the GQM plan.
- b) Some of the data in the GQM plan is based on already existing GQM plans carried out previously. The project team should review them in both cases.

After the integrated interviews, the results are documented. The assessment report includes assessment profiles, detailed findings and improvement recommendations. The GQM plan and measurement plan contain information on the measurement programme. The interviewees and interviewers together review the results, in order to collect feedback, verify findings, and make corrections if necessary.

Integrated Feedback

Integrated feedback combines the presentation of assessment results with GQM feedback sessions.

Feedback sessions are organized meetings involving members of the project team and the PROFES team. They provide an essential mechanism for supporting analysis and interpretation of the measurement results. The objective of the integrated feedback is a) to discuss the preliminary findings and results of the measurement programme, b) for the project team to interpret the data collected so far, and c) to discuss the status of improvement actions with complementary measurements. It is also possible to use the feedback sessions to evaluate PPDs using data from product and process measurements.

Feedback sessions are an integral part of goal-oriented measurement. In GQM measurement, regular feedback sessions are arranged with the application project. Integrated feedback sessions combine GQM feedback

sessions and assessment presentations, which contain a presentation of assessment results and a follow-up of improvement actions. It is important that the final assessment presentation is performed as soon as possible after completion of the interviews, so that assessment results can be presented and accepted, and improvement recommendations discussed. Case experiments during the PROFES project indicated that application projects are interested in participating in integrated sessions, since this method of implementing the PROFES improvement methodology was considered both efficient and useful for the target project and the PROFES team. Table 4.3 describes the integrated feedback session activities:

Table 4.3. Activities conducted during integrated feedback

Assessment	GQM method
Integrated feedback	
<ul style="list-style-type: none"> ▪ Implement improvement actions ▪ Follow-up of the status of software process improvements 	<ul style="list-style-type: none"> ▪ Collect measurement data
<ul style="list-style-type: none"> ▪ Feedback from assessment-related measurements ▪ Refine improvement actions 	<ul style="list-style-type: none"> ▪ Prepare and conduct feedback session ▪ Write feedback session report ▪ Update measurement programme

Two kinds of integrated feedback sessions can be arranged:

- 1) An initial feedback session in which assessment results, improvement recommendations, and project responses are discussed.

- 2) Further feedback sessions, during which collected measurements are analysed. This also includes a status presentation of improvement actions. If measurements are linked to assessment indicators, improvements in process capability can also be presented.

We recommend that the initial feedback session be held soon after the assessment results are available. The main emphasis at this session is on assessment results, findings, and recommendations, as measurement data is rarely available. GQM plans and measurement plans can be presented to the project members at once, in order to hear their immediate comments. If the company has already collected measurements that are part of the GQM plan, then those measurements can also be presented.

The subsequent feedback sessions arranged are closer to the feedback sessions recommended by the GQM approach. The main focus of the feedback sessions is to present the collected measurement data and analyse it together with the project team. An integrated feedback session differs from the standard GQM feedback session in that it also contains assessment feedback.

There are two kinds of assessment feedback that can be given:

- a) Feedback can be given on status of improvement recommendations
- b) Feedback can be given on process capability measurements

Usually, several improvement recommendations were made in the assessment, but it is not possible to measure all of them. Improvement recommendations not followed up with measurements should at least have their current status checked. Measurements collected to continuously monitor the influence of improvement recommendations and fine-tuning naturally offer more detailed information than a simple status check.

Issues related to process capability measurements are discussed in the next section, "Continuous assessment".

How to Use Continuous Assessment to Support Product-Driven Process Improvement?

In this section, we will focus on continuous assessment and discuss both its benefits and limitations. The main benefits are increased visibility of the actual software process and the ability to detect process deviations earlier than before. However, successful application of continuous assessment requires a focused improvement area, experience in goal-oriented measurement, and an adequate data collection infrastructure.

Why and When Should Continuous Assessment Be Carried Out?

Software process assessments have become commonplace in the software industry. However, assessments are sometimes too infrequent, expensive and disruptive. Therefore, there is a clear need for alternative ways to assess the current status of software processes and to monitor the implementation of improvement activities.

Experience of the practical application of continuous assessment suggests that this approach is feasible, and provides a structured framework for process capability monitoring during software development, offering new insights into measurement programme at a reasonable cost.

Expected Benefits

There are three main areas where continuous assessment is expected to bring benefits over the traditional approaches:

- Process visibility
- Detection of process deviations
- Assessment cost

Process visibility

With continuous assessment, the process implementation becomes more visible by using a reference model for software processes. This means

that it is possible to see what is being done to the software process in greater detail. This enables close monitoring of improvement activities, making it more apparent whether new practices are successfully adopted long before the usual re-assessment.

Detection of process deviations

Continuous assessment also provides a means of detecting process deviations earlier on, thus improving process implementation management in two ways. Firstly, by giving early signals of practices that are not being adopted, which indicates that people should be supported in the process adaptation. Secondly, by suggesting potential for process change, as defined processes and procedures are typically rigid. In practice, processes are dynamic, i.e. always subject to change. A better visualization of process implementation can help in identifying processes that need to be changed, or are already being changed by the people using the process. In this way, processes can become living representations of the work, rather than a row of folders collecting dust on bookshelves.

Assessment cost

Assessment costs are expected to diminish through continuous assessment. Collecting information from the software process as it becomes available reduces the time needed for interviews and document analysis during an assessment. Appropriate tooling, such as MetriFlame, can greatly support this data collection.

Prerequisites

Continuous assessment is not suitable for everybody. Successful implementation of continuous assessment requires:

- Focused improvement area
- Measurement experience
- An adequate data collection infrastructure

Focused improvement area

The reason for carrying out continuous assessment is usually to closely monitor a process either critical for overall performance or undergoing improvement. Like any kind of measurement programme, setting up a

measurement programme for continuous assessment can require substantial effort. For this reason, we generally recommend that processes for continuous assessment be carefully selected before starting.

Measurement experience

If a company has never carried out goal-oriented measurement, we recommend that work be begun with simple measurement goals while experience is gained during measurement. This measurement experience is beneficial when defining and using capability-related metrics. Further, we suggest that continuous assessment should be part of an existing measurement programme, and that continuous assessment results should be discussed in a GQM feedback session. Without any experience of interpreting measurement data, it may be difficult to exploit the results of continuous assessment and draw practical conclusions.

Adequate data collection infrastructure

The key is to integrate data collection into the work processes in such a way that it becomes a natural part of the daily work. This can be achieved in two ways: either data collection is essential for the work to be performed, for example writing an inspection report, or the work automatically leaves traces in the tools and databases of the company. When automating and integrating data collection, the cost/benefit ratio should be optimized.

Roles and Effort

A PROFES expert experienced in GQM and assessment assumes the leading role in continuous assessment. Assessment expertise can also be obtained from an external source.

When starting to apply continuous assessment the effort per process is around 30 hours if adequate generic mapping is available between processes and metrics. Creating generic mapping between a software process and metrics can take 60 hours.

Introduction to Continuous Assessment

The purpose of software process assessment is to determine the status of the software processes comparing them with a reference model such as ISO 15504 or CMM. An assessment team usually carries out a complete or overview assessment (see Figure 4.2) at infrequent intervals, perhaps biannually. A complete assessment requires significant effort, including multiple interviews and document reviews. Assessment leads to recommendations for improvement that are then prioritized and implemented over time. A focused assessment may then be performed on those processes selected for improvement. Such focused assessments naturally require fewer resources to perform, but are still conducted in a traditional manner. On the other hand, continuous assessment employs a different paradigm for conducting assessment.

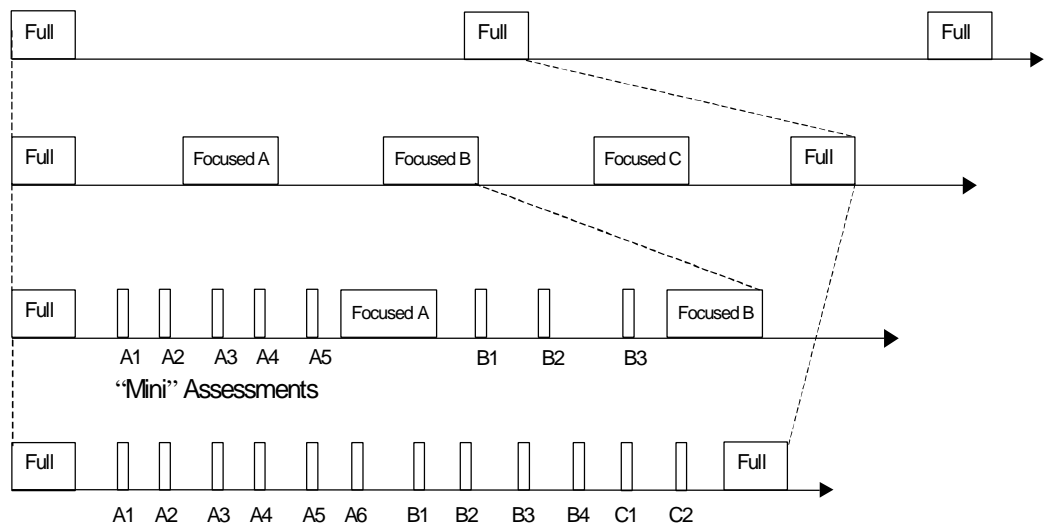


Figure 4.2. Assessment scenarios.

The idea of continuous assessment is to collect information from the software process as it becomes available during software engineering work and make “mini”-assessments (see Figure 4.2) at predefined intervals, such as project milestones. The mini-assessments can provide information for focused assessments and sometimes even replace them. It is still a good idea to make full assessments, biannually for example, and use them to have an overview on all processes.

Assessment as a Measurement Instrument

There are various ways to implement continuous assessment, for example in a process-centred development environment or through intensive data collection procedures. Our approach is to use continuous assessment as a measurement instrument that complies with the GQM paradigm, i.e. by conducting “mini”-assessments using goal-oriented measurement data (see Figure 4.3). The white areas in the GQM bar represent GQM planning, and the grey areas represent execution of the measurement programme. The solid arrows signify flows of information for measurement planning purposes, and the dotted arrows represent the flow of measurement data for capability assessment purposes.

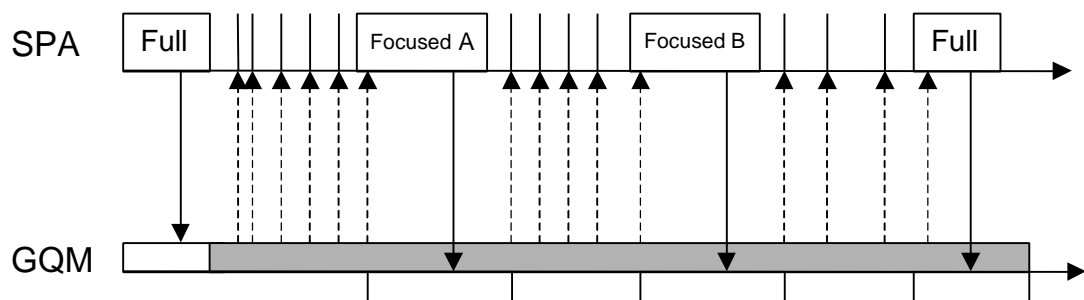


Figure 4.3. Information flow between assessment (SPA) and measurement programme (GQM).

We see process assessment results as a set of metrics for the measurement programme. Software process assessment is conducted using a specific process reference model and rules for assessment, and for calculating results. Therefore, it can be argued that assessment results are measurements, even if they are complex measurements. They can then be used in a goal-oriented measurement programme as any other measurements – to answer specific questions. In practice, this means adding a goal or subgoal to the GQM plan, for example to analyse the system test process by understanding the factors affecting process capability.

Linking Metrics to a Reference Model

A prerequisite for continuous assessment is that a mapping exists between actual project measurements and a reference model for software processes. We have chosen the forthcoming ISO 15504 standard on software process assessment as a framework for software best practice, and as a reference model for software process capability. Other models, such as CMM, are also possible but are beyond the scope of this document.

When the ISO 15504 reference model is enhanced with the assessment model defined in Part 5 of the standard, it is possible to find links between measurable objects and the ISO 15504 framework (see Figure 4.4). The assessment indicators specifically provide adequate detail for connecting process information to the framework. Process performance indicators are used to determine whether a process exists in reality.

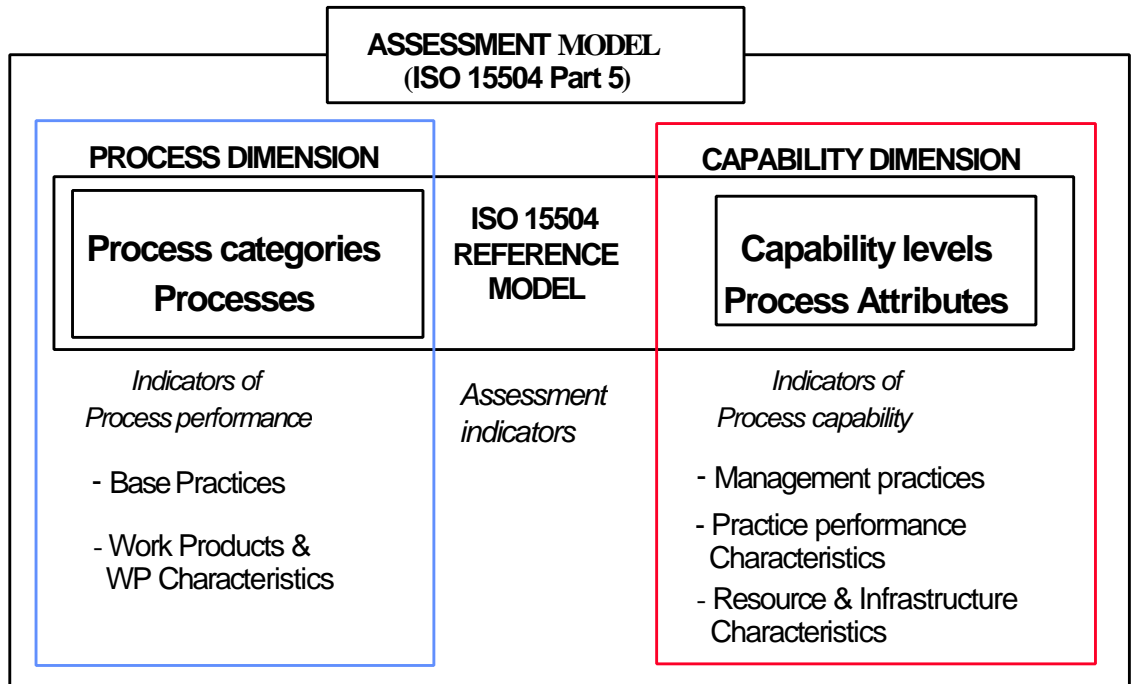


Figure 4.4. The ISO 15504 framework for software process assessment.

For example, the software design process (cf. ENG.1.3 in ISO 15504 reference model) is considered to exist if it can be determined that documents exist that specify:

- Architectural design that describes the major software components that will implement the software requirements
- Internal and external interfaces of each software component
- Detailed design that describes software units that can be built and tested
- Consistency between software requirements and software designs.

If a software design process is functioning in an organization, it should be straightforward to determine the existence of documents that satisfy the goals listed above. For example, this information can be found in a document management system that tracks the documents produced with a

specified process. A report from this system can then help an assessor to determine whether the software design process is being performed.

After determining the existence of a process, the ISO 15504 indicators can then be used to determine the capability of an existing process. Linking information from the measurement system to management practices, practice performance characteristics, resources, and infrastructure can help an assessor to determine how well the process is performed in relation to ISO 15504. For example, the performance management attribute 2.1 of ISO 15504 Level 2 can be considered as fulfilled if:

- objectives for the performance of the process will be identified, for example, schedule, cycle time, and resource usage
- Responsibility and authority for developing the process work products will be assigned
- Process performance will be managed to produce work products that meet the defined objectives.

Generally, it is more complex to use measurement data to assess process capability than using measurement data to demonstrate that processes exist. See the example on continuous assessment later in this chapter.

Adaptation and Reuse of Metrics for Continuous Assessment

The ISO 15504 reference framework and assessment model and assessment indicator set can be utilized to map and reuse measurement items related to process capability. However, for optimal results we recommend that organizations adapt their own indicator sets based on the ISO 15504 reference model, as the indicator set defined in ISO 15504 is a generic set only intended for guidance and as a starting point. Adaptation does not need to be extensive, but at least the suitability of available indicators should be ensured.

Adaptation is begun by mapping the ISO 15504 indicators to the relevant items in the organization, for example differentiating between embedded systems development and office software. With a tailored process capability indicator set, organizations can focus more specifically on the problems in their processes, and of course continue to refine the indicators for better precision and coverage in the future.

Guidelines for Continuous Assessment

This section describes a method for continuous assessment. This approach has been motivated and constrained by the requirements of the industrial application cases in the PROFES project, which aimed to ensure the practical applicability of continuous assessment.

Steps for Applying Continuous Assessment

There are six steps for applying continuous assessment. Its prerequisites are that at least one overall assessment has been made previously, and that goal-oriented measurement is being planned or revised. It is difficult to select a limited set of processes if the overall capability is not known. In practice, experience has shown that continuous assessment typically has the highest cost/benefit ratio when used to augment an existing goal-oriented measurement programme.

The six steps to apply continuous assessment are as follows:

- I. Select processes to be examined
- II. Construct or update measurement goals
- III. Define indicators for process existence and capability
- IV. Construct or update measurement plans
- V. Collect data and assess selected processes
- VI. Analyze results and do corrective actions

I. Select processes to be examined

The principle in selecting processes for continuous assessment is that only those processes that are either critical or currently being improved are included. Generally, it is worth starting with just one or two processes in order to gain experience of continuous assessment. In short, prospective processes for continuous assessment are those that are a) have already been measured, b) are being, or planned to be improved, and c) are extensively supported by tools to minimize manual data collection. The

selected processes should then be prepared for continuous assessment so that:

- A target rating is recorded for each practice, which can be the same as current rating if only monitoring is attempted. This is the starting point for systematically governing improvement activities
- Applicable sources for measurement data are defined. Examples of good data sources with potential for automatic data collection are Lotus Notes, MS-Project, any configuration management system, or any database that is used to collect project data. However, the data does not have to be always automatically collectable, although this is usually preferred.

II. Construct or update measurement goals

The measurements relating to process existence and capability are typically integrated into an existing measurement programme. Therefore, the measurement goals are updated, or new measurement goals need to be constructed to accommodate the capability-related metrics.

III. Define indicators for process existence and capability

For each selected process, the most important measurements are those indicating whether the process is performing or not, that is producing useful results and fulfilling its purpose. This is the ISO15504 process dimension. Depending on the scope chosen, the metrics related to the ISO15504 capability dimension can also be reviewed. These metrics are used to measure control, management, and improvement aspects of the process. Please note that there may be practices that are better left for assessment interviews; not everything needs to be covered automatically.

a) Process existence indicators

The ISO15504 process dimension includes base practices that are the minimum set of practices necessary to successfully perform a process. For example, the base practices for the Software Construction Process (ENG.1.4) that covers coding and unit testing in a software life cycle are: Develop software units, Develop unit verification procedures, Verify the software units, and Establish traceability. Metrics suitable for base practices are usually those that give evidence of the base practice existence, i.e. that something has been done that contributes to fulfilling the purpose of the process. Information should usually be found in the artifacts, that is the work products that are produced in a given process.

b) Process capability indicators

The ISO15504 capability dimension should also be examined for the selected processes. The ISO15504 capability dimension contains information on how well the practices are performed, and how well the process runs. Usually, going through Level 2 of the capability dimension is enough, as this is the present state of the practice. Recent SPICE assessment trials results show that only 12% of process instances (341 in total) were higher than Level 2. Naturally, higher levels can be revisited depending on target capability. Information for identifying the capability dimension can mostly be found in the project plan, project reporting documents, configuration management system, and the actual work products.

IV. Construct or update measurement plans

The definition of relevant measurements for continuous assessment does not necessarily require using a goal-oriented measurement plan with goals, questions, and associated metrics, as the ISO15504 processes form the structure for the investigation. However, an existing GQM plan is an excellent source of information. Some of the GQM measurements may also be used to facilitate software process assessment. Augmenting an existing measurement programme with process capability focus provides added value at reasonable cost. For example, it is possible to monitor process improvement activities closely and evaluate the effectiveness of the process changes.

The integration of measurement activities into the software process must be planned with care. Usually this involves at least minor changes to the process, as data must be recorded or structured in a way that is processable later. Software tools and databases are a key source for process data, but even then effort is needed to structure, convert, and extract data from various tools and databases. Some data may also be entered manually from questionnaires or checklists. Within the PROFES project, various checklists proved to be particularly useful for the continuous assessment trials. See the Tokheim example later in this section for more information on the use of checklists for continuous assessment.

V. Collect data and assess selected processes

The data for continuous assessment indicators should be collected during project implementation as part of the data collection routines agreed in the measurement plan. A spreadsheet program such as Microsoft Excel may be sufficient for data consolidation and analysis, but more sophisticated tools such as MetriFlame may be needed. The MetriFlame tool was used in the PROFES project for managing the measurement data and producing graphs for analysis in GQM feedback sessions. MetriFlame also

supports continuous assessment by providing links between GQM-based metrics and ISO15504 processes. The frequency of assessments varies, but project milestones and GQM feedback sessions are typically good candidates for timing a snapshot of process capability. Please note that for some indicators there may be measurement data, but for some indicators a quick check on the process by a competent assessor is needed, as it is not cost-efficient to automate everything.

VI. Analyze results and do corrective actions

The assessment results from the continuous assessments are discussed and interpreted in GQM feedback sessions, similar to any other measurement results prepared for the feedback sessions. After analyzing the data, specified corrective actions are taken and data collection is continued. The measurement programme also needs to be analysed critically and altered whenever appropriate.

Example: Continuous Assessment at Tokheim

The Tokheim software development centre in Bladel has produced a measurement programme for investigating the system test process of the OMEGA system. Tokheim then adapted the ISO15504 assessment indicators to suit their system test process. ISO15504-compliant BOOTSTRAP process descriptions were used to assist the alteration work. The resulting enhanced measurement plan was used in continuous assessments of the system test process of the OMEGA system. In this section, we will describe how the continuous assessments were set up and used in practice.

Continuous Assessment: Starting from Scratch

The OMEGA project started to carry out continuous assessments with a good background in measurement, but with limited assessment experience. In practice, a Tokheim employee established the continuous assessment without much prior exposure to software process assessment or measurement. However, measurement and assessment experts at Tokheim guided his work. See Table 4.4 for the effort required.

Table 4.4. Example of effort for establishing continuous assessment when starting from scratch

Activity	Total effort	Effort
BOOTSTRAP specific GQM plan:	±100 hours	
Learning BOOTSTRAP:		20 hours
Learning GQM:		20 hours
Defining goals and questions:		10 hours
Defining metrics:		20 hours
Defining checklists:		30 hours
Constructing Continuous Assessment:	±30 hours	
Investigate GQM plan:		5 hours
Comparing plans:		15 hours
Integration:		10 hours

Finding Indicators for Continuous Assessment

To establish a setting in which continuous assessments can be performed, the existing measurement programme and the existing process improvement programme had to be integrated, as shown in Figure 4.5.

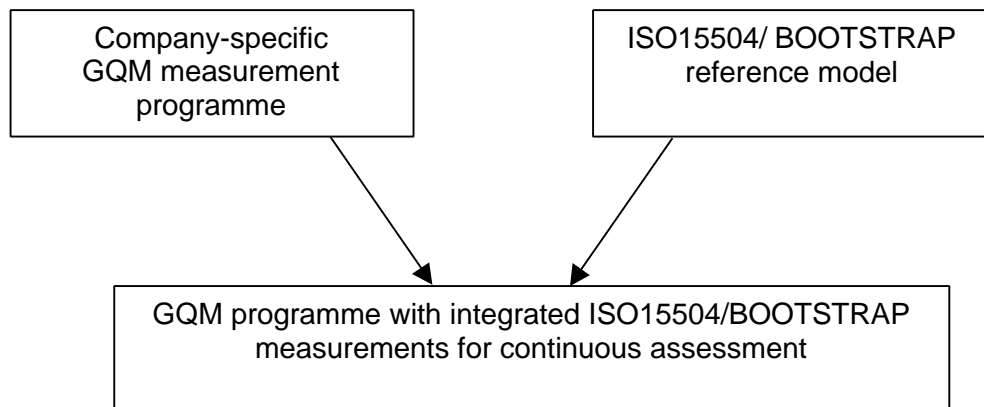


Figure 4.5. Combining GQM with ISO15504/BOOTSTRAP to address continuous assessment.

The integration towards continuous assessment was done in three steps. Firstly, the system testing process as defined by BOOTSTRAP was

investigated to find relevant goals, questions, and metrics for the process in general. It is assumed that GQM plans based on the ISO15504 reference model are generic, and therefore need to be created only once. Therefore, these generic GQM plans would be available in a future implementation of continuous assessment. Secondly, the assessment indicators for the system testing process were adapted to suit Tokheim, and specifically the OMEGA environment. This customization process is illustrated in Figure 4.6 and resulted in:

- *A direct measurement data collection plan* – a set of metrics that were related to the ISO15504/ BOOTSTRAP assessment indicators, and could be collected directly using software development tools
- *A set of document checklist items* that needed to be checked for each development document, as they were related to the ISO15504/ BOOTSTRAP work product indicators
- *A set of event-driven checklist items* that needed to be checked for a specific event that occurred (i.e. system test is started, or a defect is found), and were related to the ISO15504/ BOOTSTRAP assessment indicators.

This customization of assessment indicators was expected to be project-specific. However, a comparison with another Tokheim project indicated that the altered indicator set of OMEGA could be largely reused for other projects, even though they may be conducted in a very different application domain.

Another starting point for applying continuous assessment was the existing measurement programme. Hence, the third step of integrating assessment indicators with measurement programmes was to study the GQM plan on OMEGA system testing and decide what aspects of process capability could provide additional information to answer the questions related to the selected measurement goals. The result was a new GQM plan in which the software process capability measurements were integrated. Some of these metrics were already used in the original measurement programme, but also provided relevant information for the assessment. For example, the measurement programme measured the frequency and effort spent on regression tests and this data was collected by the testing report. These measurements could also be used as output work product indicators of the testing process.

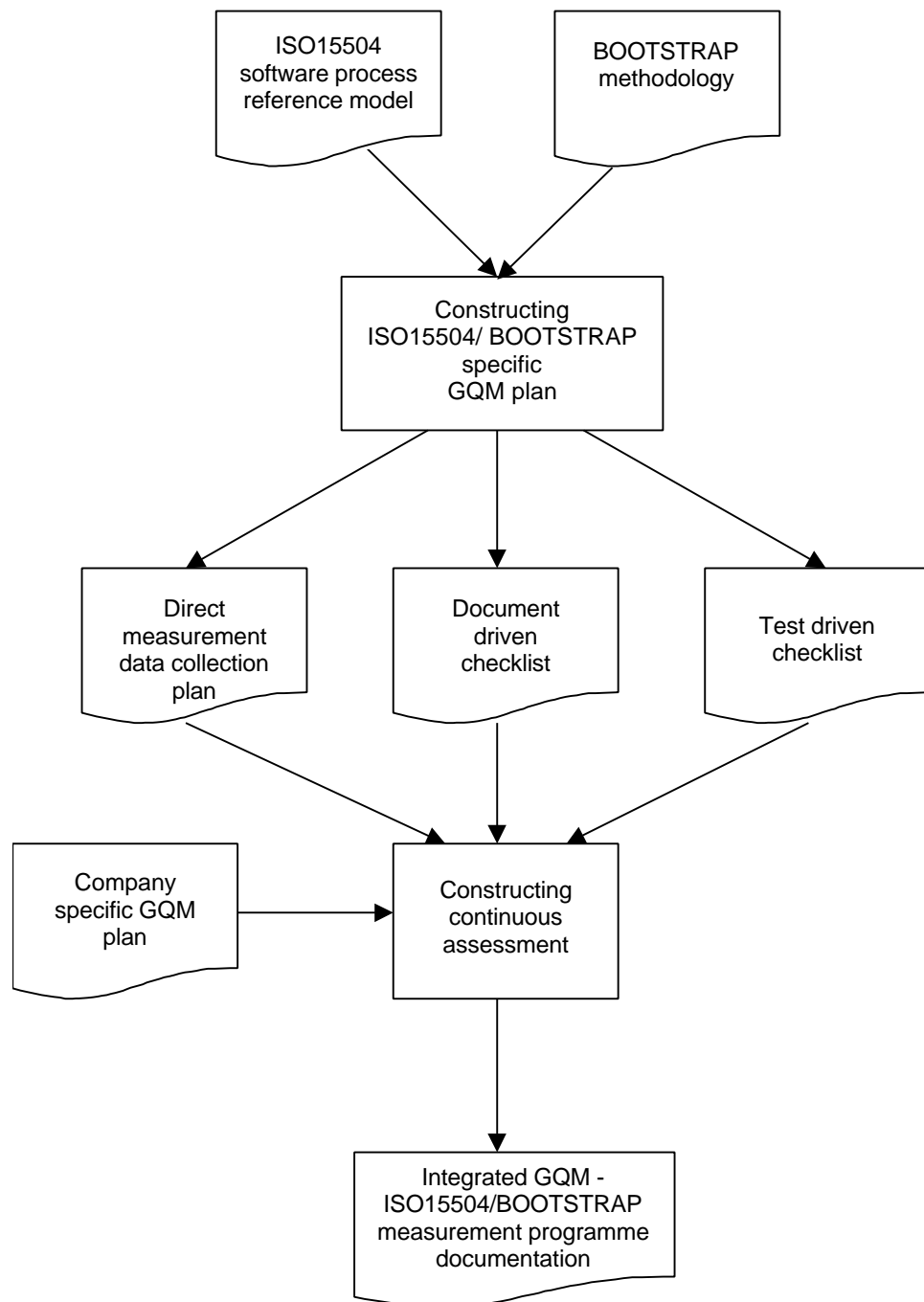


Figure 4.6. Integrating GQM measurement and ISO15504/ BOOTSTRAP assessments.

Using Measurement Data for Continuous Assessment

The approach for gathering measurement data for the measurement programme of the OMEGA project was to use multiple data collection methods, as illustrated in Figure 4.6. Some data was collected directly using the development tools. For example, the data on failure severity was retrieved from a quality problem reporting database. Some data came from interviews, but mostly the data was collected using checklists embedded into the development process. The information gained through these various methods was viewed through the GQM tree structure (see an example of GQM tree in Figure 4.7) and the integrated ISO15504 reference model. A competent assessor verified and examined the collected data, and carried out some clarifying interviews to ensure that the impression of the measurement data was correct. Then he rated the process practices, recorded findings, and generated a process rating profile. This process profile was discussed in a feedback session along with other material from the measurement programme.

Tokheim Experience of Continuous Assessment

The continuous assessment approach provides added value if it is combined with an existing measurement programme. For example, mapping project activities compared to a state-of-the-art software process reference model gives additional confidence in monitoring the OMEGA system test process. An indication of added confidence was that an existing metric was typically linked to new, capability-related questions. The continuous assessment information also provided new insights into the GQM feedback sessions. For example, the factors affecting actual process implementation became very clear to the process participants, resulting in improved process implementation. However, with an unclear focus or insufficient infrastructure for data collection, it is likely that continuous assessment would cause significant overhead. Yet we found that sufficient infrastructure for data collection does not necessarily imply state-of-the-art tooling or large overhead in manual data collection. Using checklists embedded into the process was a very effective and efficient method to collect measurement data.

How to Use Advanced Analysis Techniques in PROFES

This section gives an introduction to some important techniques for analysing and interpreting data. Furthermore, hints on how to analyse and interpret measurement data in feedback sessions are given.

In the PROFES improvement methodology, measurement is goal-oriented according to the GQM paradigm, which means that metrics are derived from goals via questions (see Figure 4.7). In contrast, the interpretation of collected data is done in a bottom-up way.

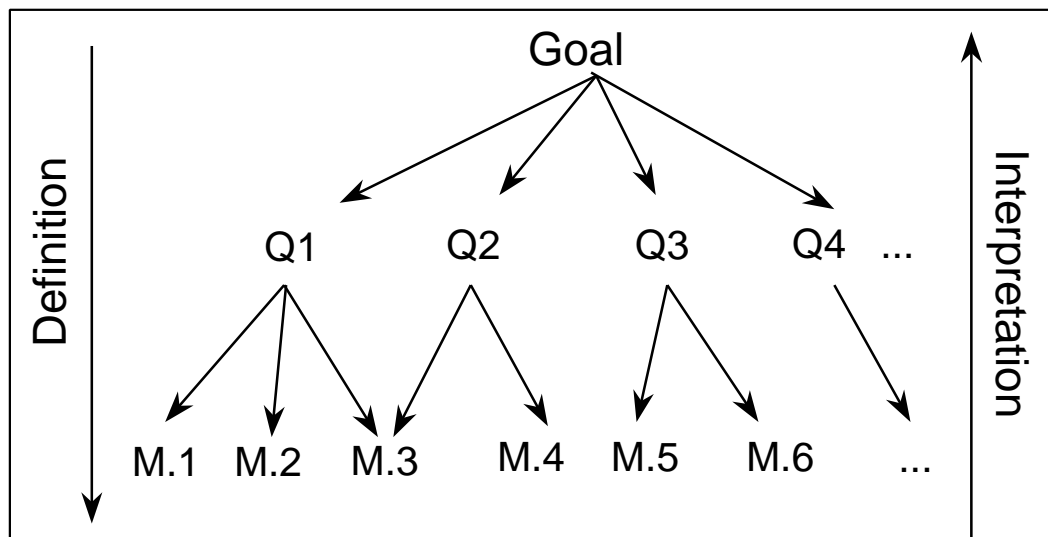


Figure 4.7: GQM Tree

Feedback sessions are the key for achieving positive results in measurement programmes. Experience has shown that it is an unnecessary risk not to have well-defined feedback sessions. In the later part of this section, we will show how feedback sessions are used to discuss and interpret the analysed data together with the engineers.

Data Mining Techniques

This section provides a brief overview of some well-known data mining techniques, which can be used to analyse the collected data. The two “high-level” primary goals of data mining in practice are prediction and description. Prediction involves using some variables or fields in a database to predict unknown or future values of other variables of interest. Description focuses on finding human-interpretable patterns describing the

data. The relative importance of prediction and description for particular data mining applications can vary considerably. There exist a variety of data mining methods. However, it is not necessary to provide a complete list of data mining techniques in this document nor explain all of them in detail. An important subset of popular techniques will be briefly introduced, but the different techniques will not be compared to each other. The data mining techniques that will be described are Optimized Set Reduction (OSR), classification trees, and the rough-set approach.

Optimized Set Reduction

The Optimized Set Reduction (OSR) approach has been developed at the University of Maryland within the framework of the TAME project. It has been applied to several software-engineering applications. It is partially based both on machine learning principles and on univariate statistics. Using a search algorithm OSR generates a collection of logical expressions that represent patterns in a data set. The goal of OSR is to identify that subset of attributes in a historical data set that provides the best characterization of the object under study. A 'good' characterization is determined by a probability distribution of the value domain that concentrates on a large number of pattern vectors in a combination of dependent variables. Each of these subsets yielding 'optimal' distributions is characterized by a set of conditions that are true for all pattern vectors in that particular subset. OSR can also be used to generate models using architectural metrics, which can be used to control software development projects.

Classification Tree

Classification and Regression Tree (CART) techniques generate partition trees, based on a historical data set describing past experience of interest. They produce interpretable classification models that help to take remedial actions based upon quantitative methods. Classification trees are used to predict the membership of cases or objects in the class of a categorical dependent variable. The classification trees try to hierarchically structure the objects under consideration by using questions and corresponding metrics data. A hierarchy of questions is presented and the final decision to be made depends on the answers to all the previous questions. Similarly, the relationship of a leaf to the tree on which it grows can be described by the hierarchy of branches, starting from the trunk and leading to the last branch from which the leaf hangs. The hierarchical nature of classification trees is one of their basic features. An example of a tree is shown in Figure 4.8.

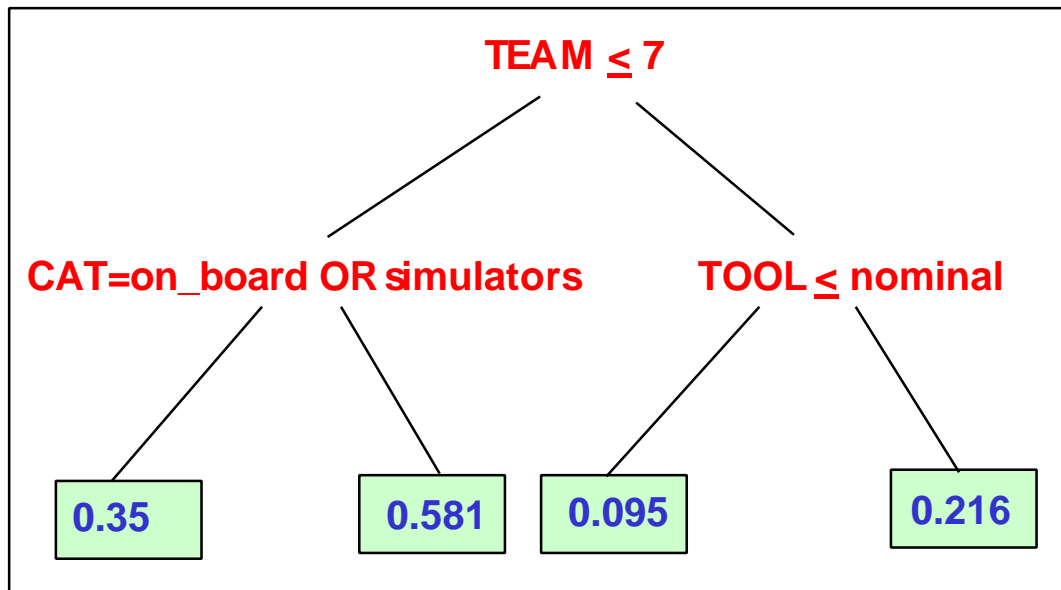


Figure 4.8: Example of CART

The example in Figure 4.8 shows the expected productivity rate (values shown in the leaves of the tree), depending on team size, category of the developed software, and tools. For example, the expected productivity rate is 0.35 when the team size is less than or equal to seven, and the category of the developed software is on board or simulator software.

Rough Sets

The Rough Sets theory is a mathematical approach dealing with uncertainty in measurement data. It was first introduced in 1981 at the Warsaw University of Technology, and has been successfully applied to data analyses in different areas. The original idea of the Rough Set theory has been expanded by adding a rule production approach.

We assume a data representation called Decision table. Rows in this table correspond to software objects, and columns correspond to attributes. For each pair (object, attribute), there is a known value called a descriptor. These descriptors have to be on an ordinal or nominal scale. All this data is summarized in the notation of an information system, which is formally defined as a 4-tuple $S = \langle U, A, V, f \rangle$ where

- U is a finite set of objects
- A is a finite set of attributes with $A = C \cup D$, $C \cap D = \emptyset$, where C is the set of condition attributes (related to dependent variables), and D is the set of decision attributes (related to independent variables).
- $V = \bigcup_{q \in Q} V_q$, where V_q is the domain of the attribute q

- $f: U \times A \rightarrow V$ with $f(x,a) \in V_a \quad \forall x \in U, \forall a \in A$, where f is called information function

Knowledge representation in form of production rules describes which conclusions can be drawn from which assumptions. Tools can be used to generate the production rules. Some examples on how these production rules may look like are presented in Table 4.5.

Table 4.5. Example for Rough Set production rules

1	$(\text{design_inspection, pbr}) \wedge (\text{relative_design_effort, design}) \Rightarrow (\text{fault_density, low}) \wedge (\text{relative_cost, low})$
2	$(\text{design_inspection, none}) \wedge (\text{code_inspection, ad-hoc}) \wedge (\text{module_test_tool, no}) \Rightarrow (\text{fault_density, high}) \wedge (\text{relative_cost, high})$
3	$(\text{module_test, high}) \wedge (\text{module_test_tool, yes}) \Rightarrow (\text{fault_density, low}) \wedge (\text{relative_cost, high})$

These production rules have the following meanings:

1. If perspective-based reading is used as a design inspection technique and more effort is spent on design than on coding, then the fault density and relative costs respectively will be low.
2. If no design inspection and no module test have been performed, and the code inspection is performed ad-hoc, then both the fault density and the relative costs will be high.
3. If the number of test cases in the module test is high and the test was supported by tools, then the fault density will be low and the relative costs will be high.

Using Feedback Sessions to Stimulate and Discuss Data Analysis

This section describes the principles of feedback sessions.

Purpose of a feedback session

Feedback sessions are organized meetings that unite project members, the measurement team, and the improvement initiative of the organization. It is an essential device for analysis and interpretation of the measurement data. The main objective of feedback sessions is to discuss the results of

the measurement programme and to derive interpretations by the project team from the data collected so far. Depending on the results and the current status of both the measurement and the software development process, immediate changes and modifications for both the software development process and the measurement process may be suggested.

Involved Roles

The following roles are involved in the planning, implementation, and analysis of feedback sessions:

- *Data Collector.* Data collectors are responsible for collecting the data. They can often explain anomalous data and check the completeness of the data, because they know the circumstances under which the data has been collected. Additionally, they can give feedback on the data collection material, for example questionnaires.
- *Manager.* Managers are individuals who can make and follow-up on decisions (e.g. department heads, project leaders, and group leaders). Therefore, it is very important to involve them in the measurement programme in general and particularly in feedback sessions.
- *Experience Engineer.* The Experience Engineer analyses the measurement data. He has a sound knowledge of statistical methods and data mining techniques, and is responsible for detecting trends and comparisons of hypotheses with the real data. If he detects some anomalous data, he has to talk with data collectors or the librarian.
- *Librarian.* The librarian is responsible for regular storing the measurement data and simple analysis. He delivers the data and simple standard analysis to the experience engineer.
- *Measurement Team.* The measurement team is responsible for checking the correct application of the GQM method. They maintain the documentation of the measurement programme and train new members in the measurement programme. Furthermore, they ensure that the goals of the GQM plan are reflected and that data is analysed in a bottom-up manner in the feedback sessions.
- *Moderator.* The moderator leads the feedback session. He presents the data and moderates the discussions. He is responsible for seeing that everyone's opinion is reflected, and that no one dominates the feedback session.
- *Viewpoint representatives:* Viewpoint representatives are individuals whose viewpoints are reflected by the GQM goals. They are respon-

sible for interpreting the presented data from their own point of view. They propose improvement suggestions, and provide feedback on the measurement programme.

Figure 4.9 shows the interplay of the roles mentioned above.

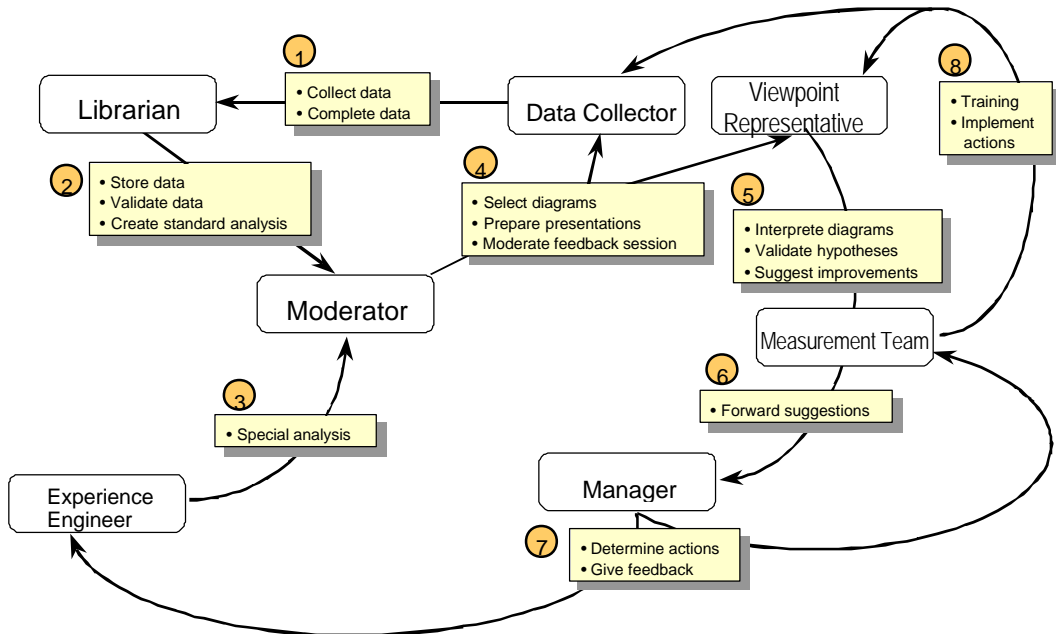


Figure 4.9: Interplay of roles in feedback sessions

Arrows indicate data flow between roles, and boxes represent the most important activities.

Preparation for a feedback session

The moderator is responsible for preparing for the feedback session. The following steps describe how the preparation can be done:

1. *Choose the main topic for the feedback session:* It is very important to concentrate on a few matters during one feedback session, as only a limited amount of time is available. Therefore, one main topic should be chosen, which will then be discussed in the feedback session.
2. *Selection of a subset of the measurement data:* Due to time and topic restrictions, not all data can be presented in one feedback session. The following criteria should be applied:

§ Questions in the QM plan

§ Number of existing and validated data points

- Number of new data points (since last presentation)
 - Degree of deviation from the hypothesis
 - Anomalous data
 - State of the development process and the product
3. *Creation of presentation slides:* The following criteria should be applied when creating slides for the feedback session:
- Diagrams must be easy to read and easy to understand for all participants
 - Related GQM questions should be mentioned in the slides
 - Number of underlying data points should be mentioned
 - The underlying hypothesis should be mentioned
 - Always use the same kind of diagram and scale for the same kind of information
 - Presented material must be coherent and not overly complex
4. *Distribution of presentation material:* Presentation material must be distributed to all participants at least a day or two before the feedback sessions.

Presentation of the data

The moderator is responsible for the implementation of the feedback session. The following steps should be followed for each slide to be shown:

1. *Explanation of the slide:*
 - Explain and describe the content
 - Explain the axes (if there are any)
 - Give some examples for the diagram

2. *Ask for interpretations:*

- Ensure that people are interpreting the presented diagrams from their own viewpoint only.

3. *Techniques for interpretation:*

- Compare the interpretations with the hypothesis in the GQM plan
- Discuss the corresponding GQM question
- Ask all participants in the feedback session for comments
- Agree on a consensus

4. *Set-up chain of arguments:*

- Identify the problem with the help of the data
- Find cause and effect with the help of interpretations
- Find solutions to the problem based on improvement suggestions

5. *Consider the impact factor:*

The moderator should check whether the impact factors mentioned during GQM interviews are (still) valid after discussion and interpretation of the collected data.

6. *Collect improvement suggestions:*

Improvement suggestions for the identified problems should be collected, forwarded to, and discussed with the people responsible.

Another very important matter is to ask for improvement suggestions for the measurement programme and measurement procedures themselves at the end of each feedback session.

Follow-up to the feedback session

The moderator and the measurement team are responsible for packaging the feedback session results. The following steps should be performed after each feedback session:

1. *Minutes.* The minutes of the feedback session should contain all the results that have been worked out during the feedback session. Furthermore, the presentation material, interpretations of the presented material, feedback on the development process and feedback on the measurement process must be included. The minutes are distributed to all participants, the librarian, and the measurement team.
2. *Open questions.* All open questions that have not been fully resolved during the feedback session are written down and distributed, so that they are available to all participants.
3. *Identified problems and improvement suggestions.* All identified problems and improvement suggestions should be forwarded to the responsible manager. He should follow up these problems and suggestions and regularly inform the measurement team of their status.

Further Reading

Basili, V.R., Caldiera G. & Rombach, H.D. 1994. Goal Question Metric Paradigm. In Marciniak, J.J, (ed.), Encyclopaedia of Software Engineering. John Wiley & Sons, Vol. 1, pp. 528-532.

Bicego, A., Khurana, M., and Kuvaja, P. 1998. BOOTSTRAP 3.0 – Software Process Assessment Methodology. In the Proceedings of the SQM '98.

ISO/IEC TR 15504-2. 1998. Information Technology – Software Process Assessment – Part 2: A Reference Model for Processes and Process Capability, Technical Report type 2. International Organization for Standardization (Ed.), Case Postale 56, CH-1211 Geneva, Switzerland.

Järvinen, J., Hamann, D. and van Solingen, R. 1999. On Integrating Assessment and Measurement: Towards Continuous Assessment of Software Engineering Processes, in the Proceedings of METRICS '99.

MetriFlame, Measurement and Feedback Tool of VTT Electronics. 1999– [online]. VTT Electronics, Finland. <http://www.ele.vtt.fi/docs/soh/metriflame/index.html>. [accessed 8.7.1999].

Paulk, M.C. et al. 1993. Capability Maturity Model for Software, Version 1.1, Software Engineering Institute, Technical Report CMU/SEI-93-TR24.

Rini van Solingen and Egon Berghout, " The goal/question/metric method, a practical method for quality improvement of software development", McGraw-Hill ISBN 007-709553-7, 1999. This book can be used as a practical guide for goal-oriented measurement programmes, including preparation of feedback material.

Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smith, Ramasamy Uthurusamy. "Advances in Knowledge Discovery and Data Mining". MIT Press, Cambridge, Massachusetts, 1996. This book is a very good introduction to various data mining techniques, which can be used for analysing the measured data.



Chapter 5

COST AND BENEFIT OF PROFES

The PROFES improvement methodology was subject to comprehensive cost/benefit analysis during pilot application in three industrial software organizations. The results of this analysis show that the PROFES improvement methodology is ready for use in industrial applications. It effectively helps to achieve important product quality improvements and additional benefits that outweigh by far the inherent costs of implementing the PROFES approach.

This chapter presents the main cost/benefit characteristics of the PROFES improvement methodology, followed by a brief introduction to the PROFES approach for cost/benefit analysis. Finally, an approach to continuous cost/benefit analysis of improvement programmes is presented. We recommend that such continuous cost/benefit analysis becomes standard practice in PROFES improvement programmes, as this facilitates the effective planning, monitoring, and control of improvement programmes.

Cost/Benefit of PROFES in a Nutshell

The PROFES cost/benefit analysis has shown that the PROFES improvement methodology:

- Can be effectively applied in industry
- Achieves product quality improvement
- Is cost-effective
- Provides benefits that outweigh the costs

The following subsections briefly outline each of these advantages.

PROFES Applicability

The high standard of applicability met by the PROFES improvement methodology is supported by experience at the three PROFES industrial pilot application sites. The pilot applications have used PROFES for more than two years, and continue to do so. Each pilot applicant has been well satisfied with the applicability of PROFES. Feedback from the pilot application sites has been used to further enhance the methodology.

The main reasons for this high standard of applicability are:

- PROFES is based on well-established and developed approaches to improvement
- PROFES integrates and combines other improvement approaches so that their particular strengths can be deployed exceptionally well
- PROFES is goal-driven and focuses all improvement activities on the organization's specific product quality goals
- PROFES employs a modular approach that can be adapted to the specific needs and characteristics of an individual software organization.

Product Quality Improvements Offered by PROFES

Experience at the three PROFES pilot application sites has shown that the PROFES improvement methodology effectively directs improvement efforts toward the achievement of important, organization-specific product quality improvements.

The most important product quality attributes for *Dräger Medical Technology* were reliability, suitability for use, and the predictability of quality, time, and cost. Several important product quality achievements were indicated: On-schedule delivery, functionality well suited to user needs, a very low number of defects in field tests, and others. A wide spectrum of process improvements was also accomplished, which was demonstrated by a rapid process capability increase to Level 3 on the BOOTSTRAP scale, thus meeting the ISO 9001 criteria.

The product-focused process improvement programme at *Ericsson Telecom R&D* in Finland focused mainly on reliability and maintainability. A particularly important quality improvement was design quality in terms of fault density. The improvements were attributed to significantly more thorough preparation of software inspections than previously, and more intense desk checking. Two BOOTSTRAP process assessments have indicated capability level improvements from below Level 2 almost to Level 3.

The product quality goals for *Tokheim*, the world market leader in systems and services for fuel stations, focused on reliability and strict cost and schedule targets. Achievements included a well-structured product architecture, better traceability and analysis of the product, as well as a very low number of defects. At the same time, the reduction in cost was better than planned and product delivery remained within planned limits.

Cost-Effectiveness of PROFES

The PROFES improvement methodology can be applied at low overhead cost. This is mainly due to the focused integration of several specialized improvement approaches. In addition, the focus on important company-specific product quality goals provides rapid progress during the improvement programme.

As cost indicator, PROFES has focused on effort of personnel. It is generally relevant, and it also is the most basic cost factor. Other cost indicators, such as financial cost, can be derived from the given effort data. This way, cost estimations can be adapted to arbitrary organisation-specific cost structures, for instance, to different salary systems. Some of the activities involved in an improvement programme might be delegated to external contractors, such as process assessors or software measurement consultants. In the case that process assessments are to be conducted in-house, there might be training and licensing costs in addition to what is described below. The usual training effort for software measurement is covered already in the respective cost models.

A typical PROFES improvement programme can demonstrate the cost-effectiveness of the PROFES improvement methodology. Table 5.1 shows the phases of such an improvement programme, and lists the total effort and time necessary for each phase. The programme is begun and goal set in the first phase. The goals can be used for focusing process assessment and a measurement programme on those areas of the software organization and its development processes that are important for achieving the required product quality. This focusing is supported by the identification of product/process dependence (PPD). PPD identification and deployment can already begin during the initial goal setting phase. It is usually followed by an early implementation of improvement actions. The measurement programme allows for monitoring and controlling the success of these improvement actions.

The typical duration of a PROFES improvement cycle is less than one year. At the end of this period, the first substantial process and product quality improvements can usually be identified. The total effort for a typical PROFES improvement programme in a project employing ten software engineers is about 6.5 person-months. The measurement programme manager and the process assessment team carry out most of the work.

The involvement of management and the software development team is particularly low, and only takes place in a focused manner at certain key points during the improvement programme. Hence, while the PROFES improvement method can be easily scaled up for much larger teams, it will require very little additional effort.

More information on the above scenario and the cost of running PROFES improvement programmes can be found in the web-based PROFES cost/benefit repository.

Table 5.1. A typical example of applying the PROFES improvement methodology, and the effort and time needed for running the improvement programme.

Phase of the improvement programme	Effort (person-months)	Time (calendar weeks)
Start up & goal setting	0.5	2
Process assessment	2.5	6
Measurement programme	2.5	40
Identification of PPDs	0.5	2
Improvement implementation	0.5	2
Total	6.5	52

The Positive Cost/Benefit Ratio of PROFES

To sum up, the cost/benefit ratio of the PROFES improvement methodology is clearly positive. This is mainly due to the following reasons:

- PROFES focuses on the rapid achievement of important company-specific product quality improvements
- The overall effort of running PROFES improvement is minimal, due to the close integration of developed improvement approaches
- PROFES effort models are available that allow for accurate planning and effective monitoring of improvement programmes

- Overall, the costs of applying PROFES are very reasonable, and are considerably outweighed by the improvements and other benefits achieved.

Naturally, each individual improvement programme has its own specific cost and benefit characteristics. As observed during the PROFES pilot applications, the general pattern of PROFES improvement programmes show that there is a low risk of improvement programme failure. This is made possible by clearly focusing on important company-specific product quality goals. Any possible problems can be identified early on and resolved quickly, due to the rapid progress and short feedback cycles implemented in the PROFES improvement methodology.

The PROFES Approach to Cost/Benefit Analysis

Cost/benefit analysis and validation of the PROFES improvement methodology began at the start of the PROFES project early in 1997. Based on the methodology's first blueprint, the validation study was planned with early involvement of the methodology users. The investigation was separated into two 15 month-periods, during which the PROFES methodology was applied in various projects at the three industrial application project of PROFES at Dräger MT-M, Ericsson Finland, and Tokheim. The projects were subject to detailed observation by researchers responsible for the validation work. Hence, the basic design of the empirical work in PROFES is a case study repeated twice and replicated three times. Figure 5.1 shows the overall structure and the main phases of the PROFES methodology validation study.

GQM was used to identify and define the validation criteria. Two overall GQM goals were defined that differ in their viewpoints:

- Goal 1: Analyse the PROFES methodology with respect to cost/benefit for the purpose of characterisation from the viewpoint of the methodology user in the context of PROFES.
- Goal 2: Analyse the PROFES methodology with respect to cost/benefit for the purpose of characterisation from the viewpoint of the methodology provider in the context of PROFES.

For each goal, questions and measures were gained by interviewing either representatives of the PROFES application projects or methodology developers. The results were defined in the form of two GQM plans, which were used to plan data collection and analysis. Figure 4 outlines their structure. It lists validation criteria and its assumed effect.

The study was conducted in parallel at each of the three PROFES application sites that were developing some kind of embedded system. The individual products have quite different characteristics. The organizations and overall software development environments also differ considerably from each other. This combination of overall similarities, i.e. the embedded systems domain, and their various differences made the PROFES application sites very interesting environments for providing valuable cost/benefit results for the PROFES improvement methodology.

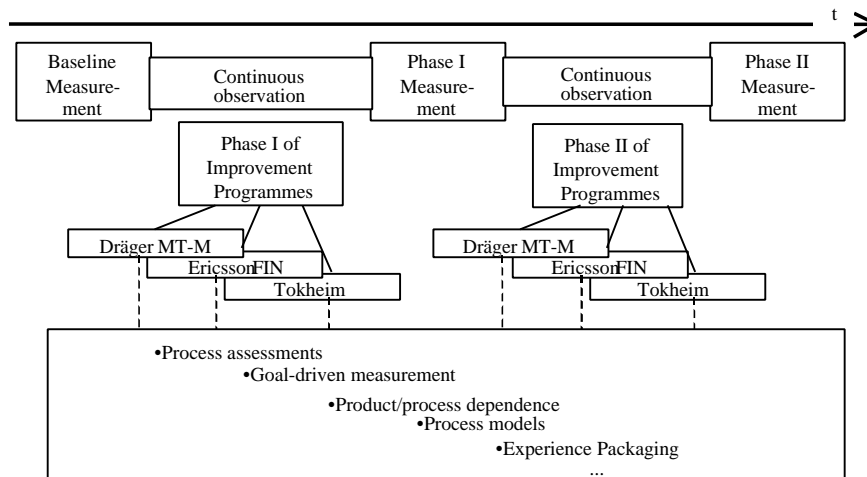


Figure 5.1. Design of the PROFES methodology validation study and its main phases.

Achievement of Product Quality Improvements

The core strategy of PROFES methodology validation is to demonstrate that application of the PROFES improvement methodology results in improvements in the developed system's product quality. This also includes aspects such as time-to-market and development costs. Validation should result in the ability to identify and explicitly document a chain of evidence, according to the pattern shown in Figure 5.2. This should be carried out according to the following principles:

- Explicitly document the complete chain of evidence, listing each element in the chain
- Clearly justify each transition in the chain from one element to the next
- For each dependent element in the chain, thoroughly identify possible alternative explanations and try to refute them
- Consider the interactive effects of multiple methodology elements that were applied concurrently
- Provide objective and quantitative evidence whenever possible, based on measurement data
- Provide qualitative and subjective evidence thoroughly.

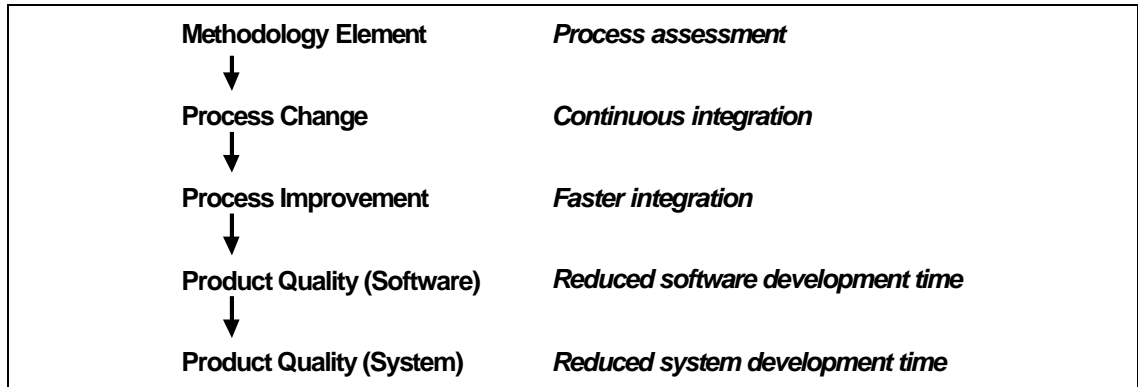


Figure 5.2. Pattern of the chain of evidence used for PROFES methodology validation with examples. (N.B. the steps process improvement and product quality (software) can be omitted in some cases.)

Table 5.2 shows a validation case example for achieving product quality through application of the PROFES improvement methodology. In this case, we recorded how the product/process dependence concept of the methodology resulted in high usability of the final system product. Compared to the causal pattern shown in Figure 5.2, the example validation case presented shows only two stages of the causal relationship. The steps *process improvement* and *product quality (software)* have been omitted. The effects of these particular process change, i.e. introduction of incremental development, on software process improvements, such as better manageability of tasks and work products, and on software product quality leading to reduced defect density, are not critical to the system product quality or usability of the final product. In other words, the impact of the particular process change on system product quality is direct. For each causal relationship in the chain of evidence, possible alternative explanations are refuted and evidence for the causal relationship is provided. The validation case shows that the product/process dependence concept in the PROFES methodology is proven.

At each of the PROFES application sites, the PROFES improvement methodology has resulted in the achievement of important product quality goals. Examples of achievements for Dräger MT-M are on-schedule delivery, with functionality easily meeting user needs, and a very low number of defects in field tests. At Ericsson, important quality improvements were achieved with regard to design quality in terms of fault density. Tokheim achieved a well-structured product architecture, better product traceability and analysability, as well as a very low number of defects during initial field tests of the product.

Table 5.2. Example case for validation of an element in the PROFES improvement methodology.

PROFES Methodology Validation Case	
Methodology Element	Product/process dependence (PPD)
<p>Process Change</p> <p>(N.B. this process change affects the system product quality directly. Intermediate effects on process and software quality are not primarily relevant and do not need to be investigated.)</p>	<p>Incremental development</p> <p>(Six-month cycles from determining requirements to system test. Implementation of core functionality in increments to allow multiple testing early on.)</p>
	<p><i>Evidence for implementation of change:</i></p> <p>The project schedule regulated cycle duration. Both product increments achieved to date were fully operational for conducting user test; therefore, primary functionality was present.</p>
	<p><i>Causal relationship:</i></p> <p>Usage of PPDs during project planning was emphasized by the PROFES project. Project management emphasized the identification of product quality goals and identified measures that were likely to help achieving these goals. In this case, quality and development schedule goals, together with the fact that many aspects of the project would be new to the team and the organization resulted in the decision to implement a incremental development process.</p>
	<p><i>Possible alternative explanations:</i></p> <ol style="list-style-type: none"> 1. Change due to process assessment 2. Not actually a change but standard development practice
	<p><i>Refutation of alternative explanations:</i></p> <p>ad 1.: First process assessment took place after the decision on incremental development was taken.</p> <p>ad 2.: None of the previous projects addressed incremental development.</p>
<p>System Product Quality</p>	<p>Good product usability</p> <ul style="list-style-type: none"> • Layouts of screen and control devices are attractive and user-friendly • Handling of the product is user-friendly
	<p><i>Evidence for achievement of software product quality:</i></p> <p>Usability of the first product increment was not satisfactory. The second increment showed good usability (according to similar statements from product marketing and from the development team).</p> <p>Engineers report that there was enough time to evaluate product usability and improve it after evaluation.</p>
	<p><i>Causal relationship:</i></p> <p>Incremental development resulted in a good user interface.</p>
	<p><i>Possible alternative explanations:</i></p> <ol style="list-style-type: none"> 1. Usability requirements were easy to implement 2. Usability only due to new hardware features
	<p><i>Refutation of alternative explanations:</i></p> <p>ad 1.: The product is of a totally new type, new user interface hardware was used, the user interface is much more complex than in previous products.</p> <p>ad 2.: The first increment's usability with the same hardware was not fully satisfactory.</p>

Non-Product-Related Improvements

The benefits of applying the PROFES improvement methodology are measured using various benefit criteria, listed in Table 5.3. They show that users of the PROFES improvement methodology, such as industrial software organizations, expect that:

- Product improvements can be achieved
- Various types of process improvement are achieved
- Improvement programmes run effectively, i.e. that they can be adapted to meet the specific needs and characteristics of the software organization, and that management resources are used efficiently
- Team knowledge and awareness with regard to various software development aspects increases
- Team building and organizational culture are supported.

The methodology provider's viewpoint provides further validation criteria, such as the quality of the methodology's documentation.

The multi-faceted definition of benefit provided by the application of the PROFES methodology has brought forward various observations from the three pilot application sites after a year of applying the PROFES improvement methodology were:

- Enhanced definitions of software development processes. Already at the GQM measurement programme-planning phase, a need for more detailed or updated software process definitions became apparent. For instance, the testing process in one project was re-designed and defined with more detail than before. Later on in the project, it became obvious that this early process change was beneficial for testing.
- Knowledge of software and system. GQM measurement has increased the project teams' knowledge of software and system, resulting in even better informed technical work and decision making. For instance, GQM measurement in one of the projects allowed for improved classification and better-informed management of change requests.
- Fine-tuning of improvement actions. During the GQM feedback sessions, previous improvement actions were fine-tuned by the software engineers in order to improve their efficiency. For instance, the way in which inspection meetings were organized in one of the projects was re-designed due to findings from measurement data analysis.

Table 5.3. PROFES validation criteria and expected influencing factors.

<i>Methodology User Viewpoint</i>	<i>Methodology Provider Viewpoint</i>
<p>Product Improvements Achievement of product quality goals</p> <p>Process Improvements Standardization of work practices Focusing of process definition Improvement of work practices Improvement of efficiency of work practices Reduced risk of failure</p> <p>Systematic Improvement Reduced risk of failure Focused improvement actions Integrated business, product, and process issues Adaptability Efficient management involvement Compatibility with quality awards</p> <p>Findings, Awareness, Understanding Knowledge of software and system Awareness of software development capabilities Awareness of critical software development issues Awareness of necessity for improvement New findings</p> <p>Team Building & Organizational Culture Contribution to group synergy Awareness of necessity for improvement</p> <p><u>Possible Influencing Factors</u> Level of software organization development Infrastructure of the software organization Other ongoing improvement initiatives Project management's awareness of the improvement methodology Top management's expectations for the improvement programme</p>	<p>Product Improvements Product quality improvements</p> <p>Process Improvements Process definition Process consistence Process stability</p> <p>Methodology characteristics Domain-specific for embedded systems development Customer viewpoint Quality and improvement awareness</p> <p>Methodology definition and support Coverage of methodology (roles, phases, activities) Assistance for methodology (processes, guidelines) Documentation for methodology Tool support for methodology</p> <p><u>Possible Influencing Factors</u> Size of measurement programme Level of software organization development Software organization infrastructure Other ongoing improvement initiatives Organizational culture: Management commitment for the improvement programme Organizational culture: Improvement attitude towards the software project Degree at which quality improvement is integrated with regular software development activities</p>

Similar information has been provided on later stages of the GQM measurement programmes, ISO 15504 process assessments, software engineering experience management, and other parts of the PROFES methodology. A presentation of the detailed results from the benefits investigation would be beyond the scope of this section. The results provide a detailed view of the application of the PROFES improvement methodology.

Cost-Effectiveness of the Improvement Methodology

The third type of methodology validation criteria in PROFES is cost-effectiveness. The GQM interviews for planning the evaluation work have resulted in the following criteria for cost-effectiveness:

- Overall effort required for the improvement programme
- Effort for the improvement programme by key personnel: managers, software engineers, improvement team, and external consultants.
- Effort required for adapting the improvement methodology when setting up the improvement programme.

The related measurements have provided detailed data on the effort required for implementing BOOTSTRAP process assessments and GQM measurement programmes. It involves the number of hours spent by each improvement programme participant on each activity of the respective method. Examples of effort models in BOOTSTRAP assessments and GQM measurement programmes are shown in Appendix 5.

During the second phase of PROFES, process assessments and measurement-related activities were conducted in an integrated manner. The effort data from these activities allows for investigation of possible synergy effects between the two techniques. In general, activity and role-based effort measurement in the improvement programme made its activities more apparent to the participants. This resulted in several clarifications made to the improvement technique definitions.

Effort measurement in the PROFES improvement programmes has shown that the overall effort for product-focused process improvement is both reasonable and acceptable to an average software organization. The modularity of the PROFES improvement methodology allows for adaptation of improvement programme activities to suit the resources available in a certain software organization. In particular, the explicit focus on relevant product quality goals, which is the core characteristic of the PROFES improvement methodology, assures that the selected improvement actions are highly effective and reduce unnecessary overhead effort during improvement programmes.

Continuous Cost/Benefit Analysis during Improvement Programmes

Associates of improvement programmes can take advantage of the cost/benefit work carried out in PROFES. The continuous cost/benefit analysis performed in PROFES offers an effective means for:

- Monitoring and controlling improvement programmes
- Achieving sustained commitment for improvement programmes

Both can be carried out with very little overhead effort, which is easily justified by assuring the success of the improvement programme.

The continuous cost/benefit analysis of improvement programmes includes the following stages:

- Explicit goal setting for product quality and process performance
- Time & effort measurement of activities in the improvement programme
- Benefit assessment at regular intervals during the improvement programme
- Joint cost/benefit analysis of all participants in an improvement programme
- Dissemination of cost/benefit results and follow-up actions

Experience of continuous cost/benefit analysis with PROFES shows that this approach is very feasible and effective. Several support tools and guidelines have been developed. They will be introduced in the following sections.

Goal Setting for Improvement Programmes

The PROFES improvement methodology emphasizes the importance of explicitly defined product and process improvement goals. Achievement of these goals is the main reason for the success of the improvement programme, and act as a basis for analysing the benefits of the improvement programme.

An improvement programme may have additional goals beyond the product and process improvement goals. Such other goals may be the fostering of cultural exchanges within the organization, e.g. establishing a higher level of co-operation between team members or achieving a quality management system certificate such as ISO 9000.

All goals and objectives imposed upon an improvement programme should be stated explicitly. They provide the basis for evaluating the improvement programme's successes and benefits. Table 5.4 lists a collection of possible goals that can serve as a checklist for goal identification.

Table 5.4. Different kinds of goals in an improvement programme.

Goal	Examples
Product quality	<ul style="list-style-type: none"> • Reliability • Maintainability • Functionality
Project performance ¹	<ul style="list-style-type: none"> • Cost of software development • Time to market
Process adherence	<ul style="list-style-type: none"> • All project team members apply the same defined process • Quality management policies are applied to the entire team
Process changes	<ul style="list-style-type: none"> • Existing software development processes are enhanced gradually • New software engineering technologies are introduced
Process management	<ul style="list-style-type: none"> • More accurate project planning • Monitoring of project progress
Team building	<ul style="list-style-type: none"> • Every team member actively contributes to problem-solving • Good interaction between team members during the development processes
Knowledge management	<ul style="list-style-type: none"> • The limited resources of senior engineers are used efficiently • Work practices and lessons learnt are recorded and disseminated

¹ Please note that in PROFES project performance, goals are categorized under the concept of product quality goals. The reason is that both kinds of goals are equally important for motivating the set-up of improvement programmes. In addition, project performance goals can also be indirectly attributed to the final product: Development cost *of the product*, and market access *of the product*.

Time and Effort Measurement of Improvement Programmes

The time and effort spent on software engineering activities are basic measurements that are collected in most software organizations. We recommend that the following time and effort measurements are also made for the activities involved in improvement programmes:

- Demonstrate that the improvement programme is cost-effective
- Identify opportunities for further enhancement of improvement programme activities
- Identify and understand possible issues early on with an ongoing improvement programme in order to take rapid corrective action
- Focus the effort available on those activities that are most important
- Facilitate the planning of future improvement programmes.

PROFES has prepared forms that support the collection of effort data for improvement programme activities. These forms are introduced in Appendix 2. Table 5.5 shows their basic design. The underlying principle is to collect effort per activity and the role involved in the activity. Annotation of the days on which the reported effort has been made allows the duration of activities to be tracked in terms of calendar days.

A prerequisite for such role and activity-based effort collection is the design of an appropriate process and role model. Table 5.1 shows the process and role models recommended for BOOTSTRAP process assessments, GQM measurement programmes, process modelling, and PPD development. The effort collection forms presented in Appendix 2 focus on process assessments and measurement programmes, as these are usually those activities within the PROFES improvement programmes that last longest and consume most of the effort. Therefore, effort collection for these activities already provides a good overview of the approximate total effort for an improvement programme.

The effort collection and analysis processes within an improvement programme should proceed according to the following steps:

- Adapt the process and role models of the improvement programme and alter the effort collection questionnaires
- Design and implement a measurement database
- Conduct a briefing and kick-off meeting with all persons actively involved in the effort collection.

- Collect the effort data, validate it, achieve any necessary additional clarification, and code the data into the measurement database
- As soon as key activities of the improvement programme are completed, summarize the effort figures for these activities
- Provide the improvement programme team with the measurement results, in order to get feedback on the validity of the effort figures
- Analyse the results in order to identify any possible issues with the improvement programme
- Package the final results as a baseline for the planning of future improvement programmes. If necessary, adjust past planning baselines based on the new figures.

Usually, it is sufficient for the effort collection to be, conducted by the key actors of the improvement programme such as the measurement programme manager and the lead assessor of the process. These persons can acquire any necessary effort data from the other persons involved and fill in the data collection forms.

Table 5.5. Basic design of effort reporting forms on activities of PROFES improvement programmes.

Activity:				
Date of reported day:				
Name of data provider:				
Time (from / to)	Activity	Roles involved	Number of people per role	Duration of activity per role

Table 5.6. Process and role models of major elements in the PROFES improvement methodology.

Process Assessments	
<p>Activities</p> <ul style="list-style-type: none"> • Preparation • Opening briefing • Global site assessment (SPU) • Project assessment (project) • Evaluation • Project assessment review • On-site final meeting • Prepare assessment report • Review assessment report • Other (please explain briefly) 	<p>Roles</p> <ul style="list-style-type: none"> • Lead assessor • Assessor • Manager • Software engineer • Facilitator • Other
GQM Measurement	
<p>Activities</p> <ul style="list-style-type: none"> • Prepare measurement programme • Identify and define GQM goals • Prepare and conduct GQM interviews • Develop GQM plan • Develop measurement plan • Perform data collection • Perform data analysis and interpretation • Package experience 	<p>Roles</p> <ul style="list-style-type: none"> • GQM Expert • Facilitator • Project management • Software engineer
Process Modelling and PPD Development	
<p>Activities</p> <ul style="list-style-type: none"> • Preparation and pre-study • Interviews and modelling • Review and approval 	<p>Roles</p> <ul style="list-style-type: none"> • Modeller • Manager • Software engineer • Facilitator • Other

Benefit Assessment of Improvement Programmes

The benefits of an improvement programme should be assessed based on an initial baseline characterization of the software organization and through repeated identification of achievements at regular points during an improvement programme. Identifying achievements can hardly be supported by standardized questionnaires, as the achievements of interest may vary greatly between different software organizations and improvement programmes. The relevant information sources can also be very different, ranging from observations via interviews to the analysis of measurement data. However, it is possible to provide generic checklists that assist the organization-specific assessment of benefits of improvement programmes.

The main points in an improvement programme when information on possible benefits and achievements should be collected are:

- At the beginning of the improvement programme, i.e. identification of the baseline situation
- After the completion of process assessments
- At the end of the planning phase of a GQM measurement programme
- After GQM feedback sessions
- After the implementation of improvement actions, such as changes in the software development process or the introduction of new technologies
- At major milestones in the software or system development process, e.g. after the completion of product increments or at the end of a field test

Table 5.7 contains a checklist of aspects of improvement programmes that may be relevant for benefit assessment. They were identified in the industrial PROFES application projects using the Goal/Question/Metric (GQM) approach. It might be worthwhile to conduct a similar, GQM-based identification of relevant cost/benefit criteria at the beginning of every improvement programme. The criteria should be acquired from the improvement programme sponsors and associates.

Table 5.7. A checklist of important benefit criteria in improvement programmes.

Product Improvements

Have the initially set product quality goals been achieved?

Are there additional product quality goals that have been achieved?

What are the assumed root causes of the achievements?

Process Improvements

Has the performance of some work practices become more consistent between the development team members?

Have process definitions or process models become updated, refined, or enhanced?

Have more effective work practices been established?

Have project risks become less likely?

Systematic Improvement

Has the risk of the improvement programme failing been reduced?

Have improvement actions been identified and implemented in a more focused and systematic manner?

Have business, product, and process issues been considered during improvement planning?

To what extent has management been involved into the improvement programme?

Findings, Awareness, Understanding

Has the team's knowledge of software and system been increased?

Has the team's awareness of the necessity for improvement been increased?

Team Building & Organizational Culture

Has interaction between team members been improved?

Joint Cost/Benefit Analysis

Cost/benefit analysis of an improvement programme should be performed on a bi-annual basis. It should be carried out as a joint effort of the managers, sponsors, and main associates of the improvement programme. The results of the cost/benefit analysis provide the basis for sustaining commitment from the sponsors and for extending or re-focusing the improvement programme.

Cost/benefit analysis is usually carried out based on qualitative data. It involves a review of the achieved cost and benefit information and the subjective assessment of the cost/benefit ratio. The basic question in cost/benefit analysis is whether the benefits achieved are worth the cost? This assessment is typically qualitative in nature, as most of the benefit information is qualitative. However, quantitative benefit assessment might

nevertheless be possible in many cases. It can be achieved by estimating the financial value of the effort spent and the improvements achieved.

Cost/benefit analysis can always provide feedback on the existing cost/benefit criteria that may lead to updates or refinements of these criteria. Therefore, cost/benefit analysis can always be adapted to changing requirements and interests for the assessment of the improvement programme.

Dissemination and Follow-Up of Cost/Benefit Results

The objective of the dissemination of cost/benefit results is to spread the “good news” of success and achievement in the improvement programme. This is primarily important for sustaining motivation, support, and commitment for the improvement programme from its participants and sponsors. Furthermore, dissemination information beyond the borders of the current improvement programme, i.e. to top management and to other projects or organizational units in the company, can help extend the improvement programme and increase the good reputation of the sponsoring organization.

Possible follow-up activities to cost/benefit analysis might be corrective actions for the improvement programme or changing an improvement programme's focus. Corrective actions might be necessary when cost/benefit analysis shows that past measures were not as effective as expected, or when progress is too time and effort-consuming. A shift of focus can result in the following actions:

- Definition of new improvement goals
- Selection of new improvement actions, i.e., process changes
- Launching of new analysis activities within the improvement programme, for example setting up a new measurement programme or conducting another project assessment

Each of these actions may result in the improvement programme being extended to other projects or organizational units in the organization. This may allow the ongoing improvement programme to benefit from experience and achievements in these other areas, and vice versa.

Further Reading

PROFES Project 1999, "*PROFES cost/benefit repository*".
<http://www.iese.fhg.de/Profes>.



Chapter 6

ESTABLISHING TOOL SUPPORT FOR PROFES

Successful adoption of any process improvement methodology often requires proper tool support. This is especially true in the case of PROFES, where several independent methodologies are used together and where the amount of data to be collected and analysed is significant. The purpose of this section is to define guidelines for establishing a PROFES tool environment.

PROFES Tool Environment

The PROFES improvement methodology includes six phases and twelve steps during which a considerable amount of data is produced. Therefore it is obvious that proper tool support is necessary for successful adoption of the PROFES methodology. However, the type and amount of data produced varies significantly between the different PROFES phases, which makes it difficult to handle all the data with a single tool. This emphasizes the need for a PROFES tool environment in which different tools can be used together so that each tool offers support to certain part(s) of the PROFES improvement cycle. Each of these tools should also utilize a logical PROFES repository in order to make information sharing between activities more efficient. The conceptual PROFES tool environment is shown in Figure 6.1.

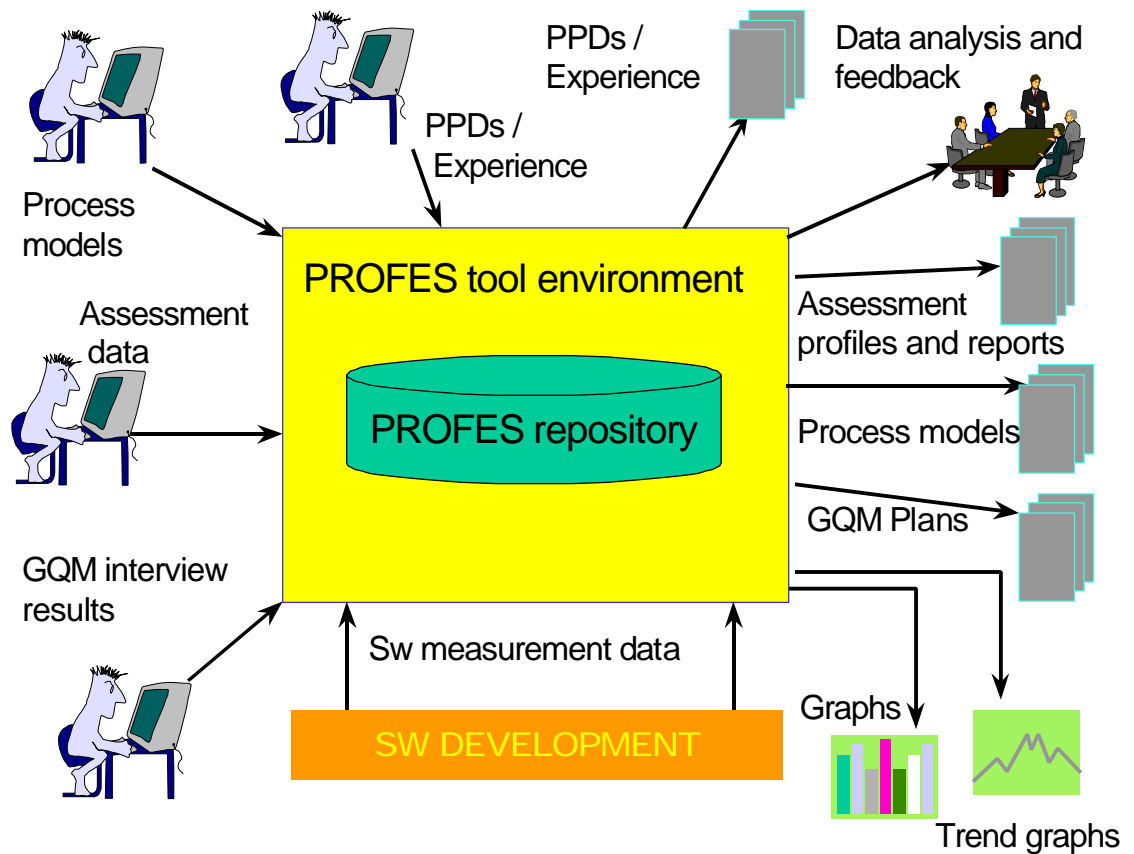


Figure 6.1. Conceptual PROFES repository and the tool environment

The above diagram also summarizes the main input and output of the PROFES methodology. Although the methodology itself includes numerous tasks and subactivities, most of the data is produced during the main background activities, which are as follows: software process assessment, GQM planning, software measurements including data collection, analysis and presentation, PPD (Product-Process Dependency) modelling, experience packaging, and process modelling. Therefore the most important functional requirements for supporting tools can be acquired by analysing these main activities.

Activities that Require Tool Support

Software process assessment includes three tasks in which a large amount of data is handled, and so appropriate tool support is particularly useful. Firstly, a specific tool is necessary for recording detailed findings and results from interviews and documentation analysis. Moreover, with the help of specific tool(s) it is easy to return to previous notes and evaluations to update them in the light of newly collected information. Secondly, rating a process usually involves a lot of calculation, which is best supported with a tool if a rating mechanism is defined in the assessment methodology. A tool is also necessary for presenting results in

the form of profiles, such as capability level profiles and attribute profiles. Thirdly, monitoring improvement action results is necessary for organizations using the PROFES methodology. One way of monitoring improvement results is to monitor the process capability trend. This monitoring task is simple if there is a specific tool in use to support longitudinal analysis of data collected from several assessments.

In the PROFES project, a tool called Bootsample was used to support the process assessment activity. Bootsample was developed by the Bootstrap Institute for use as a support tool for assessors during Bootstrap 3.0 evaluations. Bootstrap 3.0 is an ISO 15504¹-compliant process assessment and improvement methodology that provides the framework for characterizing and assessing software processes. Bootsample meets all the requirements discussed in the previous section, and offers the following additional functionality:

- Collection of information related to assessment itself and the target organization,
- Classification of the assessed organization, software producing unit, and project for the purpose of benchmarking,
- Printing of both Bootstrap and SPICE reports.

The evaluation window of the tool is shown in Figure 6.2. The Bootsample is explained in more detail in Appendix 5.

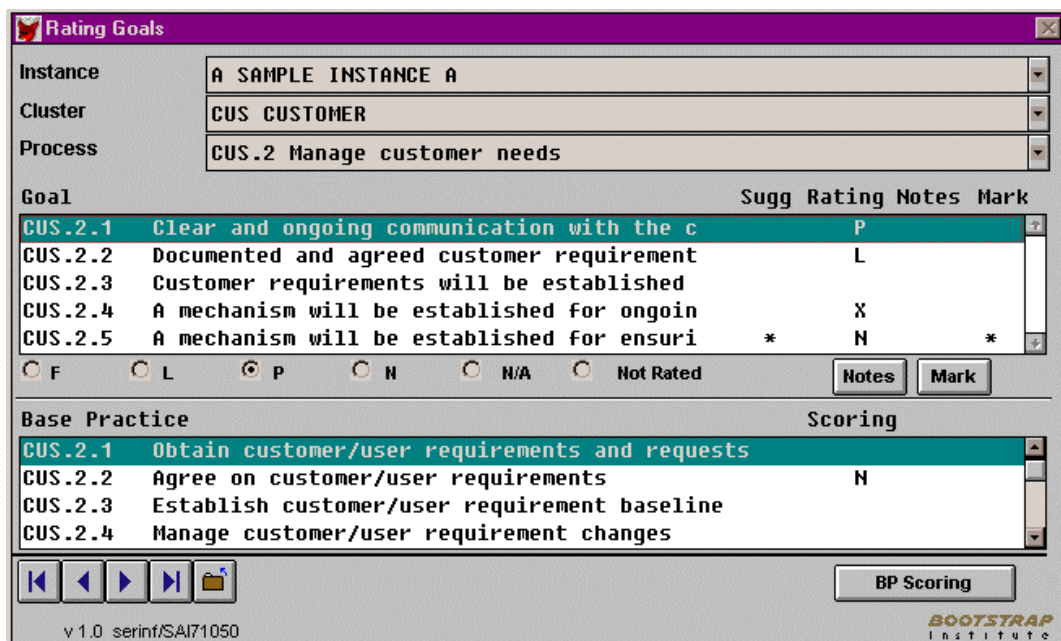


Figure 6.2 The evaluation window of the Bootsample assessment tool

¹ ISO 15504 is also known as SPICE. The assessment method used during the PROFES project is Bootstrap.

GQM planning is an activity that does not usually involve a large amount of data, and can be successfully supported by using an ordinary word processor, for example. However, a specific tool can effectively and efficiently support the GQM definition process, providing editing facilities for developing GQM abstraction sheets and GQM plans. A GQM editor can be useful in providing assistance on how to proceed, and it can also ensure that goals are defined in a methodologically correct manner.

The GQMAspect² tool was used during the PROFES project to support the planning phase of GQM-based measurement programmes. The functionality of the tool focuses on the definition and maintenance of GQM plans. The tool provides the following support for GQM planning activities:

- Templates for the definition of GQM goals,
- Templates for the definition of GQM abstraction sheets and GQM plans,
- Editing functions for constructing GQM abstraction sheets and GQM plans, involving the reuse of already existing GQM documentation,
- Automatic generation of GQM plans from GQM abstraction sheets and vice-versa.

The abstraction sheet window of the GQMAspect tool is shown in Figure 6.3. The tool is discussed more extensively in Appendix 5.

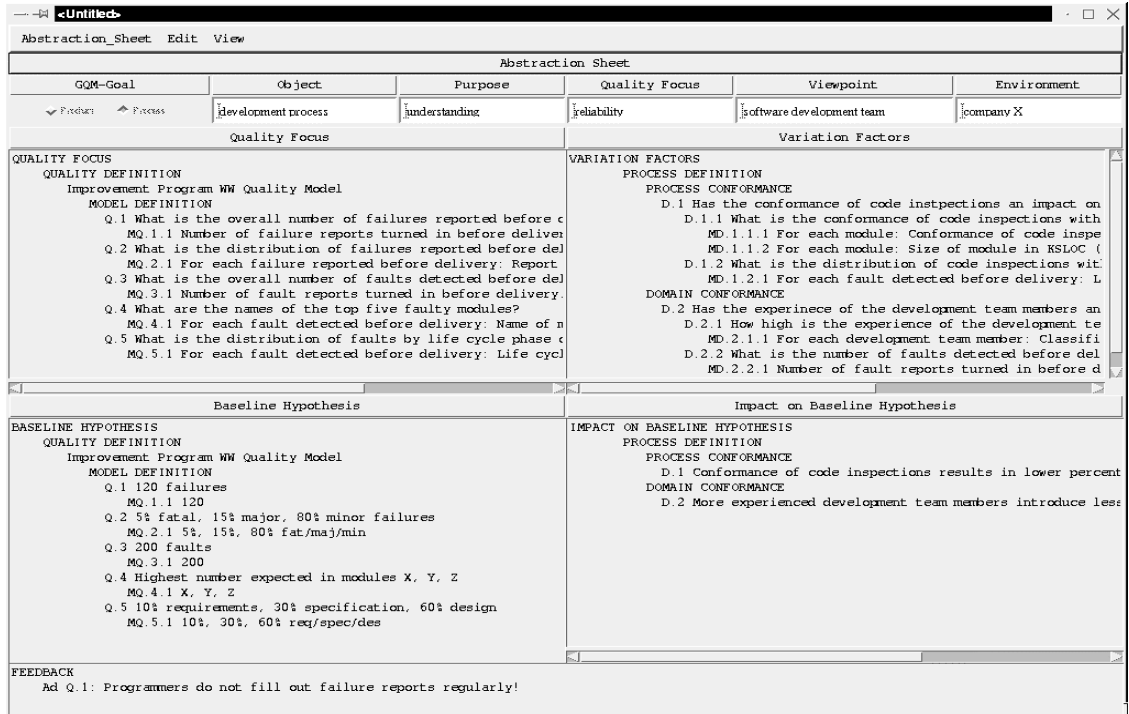


Figure 6.3 The abstraction sheet window of GQMAspect

² Developed and owned by Fraunhofer IESE

Measurement data collection, analysis, and presentation can hardly be done without tool support. Measurement data can be collected both manually and automatically. We cannot claim that tool support for data collection will provide invaluable support to a measurement programme: data collection should not overload daily routine work. Today, large amount of data can be collected automatically by software engineering tools. Tools for data analysis and presentation are necessary for collecting basic measurement data, analysing it according to defined metrics, and for producing pie charts, histograms, control charts etc.

In the PROFES project, we used a tool called MetriFlame³ for measurement tasks in the application projects. The tool provides support for goal-oriented measurement programmes that collect measurement data, define and calculate metrics, and display the results in different formats for analysis sessions. For example, the MetriFlame window for GQM plan formulation is shown in Figure 6.4.

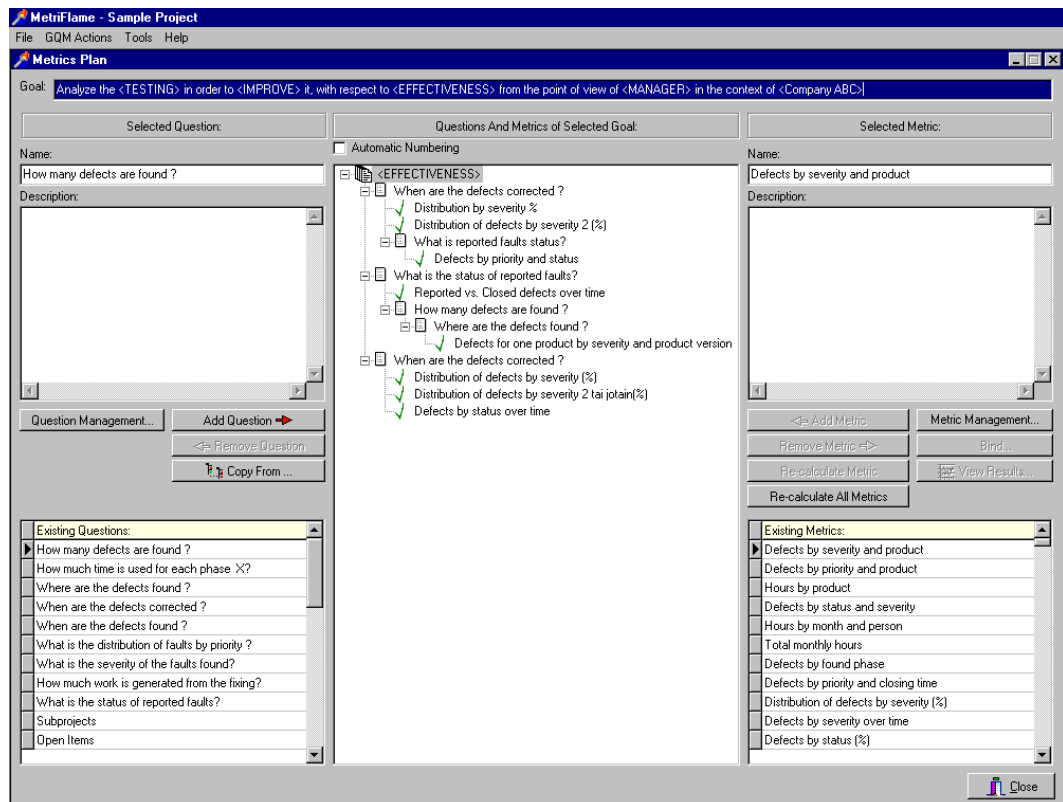


Figure 6.4. The MetriFlame GQM plan formulation window

³ Developed and owned by VTT Electronics

The functionality of the tool includes:

- Importing data from data sources, such as Lotus Notes, MS Project, testing tools, defect databases, effort-tracking databases etc.,
- Managing projects and GQM plans,
- Calculating various characteristics for the imported data,
- Visualizing the results of calculations according to user preferences,
- Following the trends of measurement results, and
- Automatic updating of measurement results with the latest data.

An example of results produced by MetriFlame is shown in Figure 6.5. MetriFlame is introduced in more detail in Appendix 5.

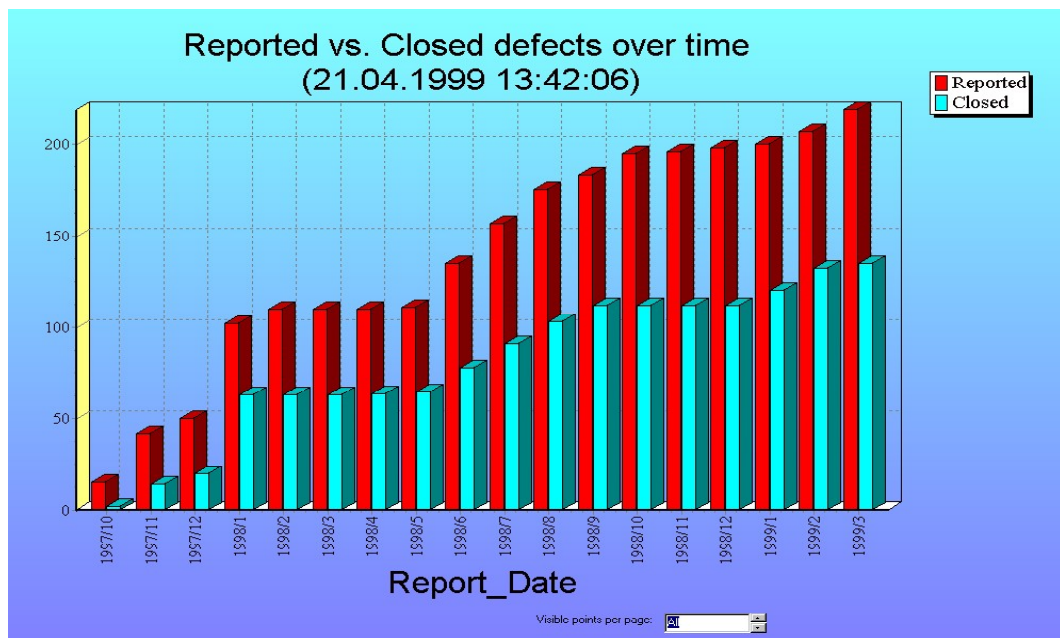


Figure 6.5. Graphic presentation of results using the MetriFlame tool

PPD modelling is a core activity of the PROFES methodology, as product quality-driven process improvement relies heavily on the availability of suitable Product-Process Dependency models (PPD models). Since a valid PPD model is a valuable asset for a software organization, it should be packaged and stored for future reuse. Packaging implies that a PPD model is augmented with additional information about the context in which it was successfully applied. Several different tools including word-processors and spreadsheets can initially be used to store PPD models. However, a specific tool is necessary to successfully manage a large PPD-repository.

As the PPD-concept is a new part of the methodology, new templates and a repository were developed during the PROFES project. Both templates and repository can be used to construct company-specific PPD models. An example of a PPD model using the PPD template is shown in Figure 6.6.

PPD-EP.1		
Product Quality	Reliability	
Process	ENG.6 Software Implementation and Testing	
Practice	Software Code Reading	
Viewpoint	Software Engineer	
Environment	Company A, Project AA	
Status	Validated	
Context		
CF.1	Code size (LOC)	<200, 200-500, >500
CF.2	Experience of project team	Low average high
CF.3	Overall time pressure	Low average high
CF.4	Preparation time (h)	<1, 1-4 , >4

Figure 6.6. Example of a PPD-model template

Experience packaging, as well as packaging of PPD models, is the responsibility of the experience factory organization. An experience base containing PPD experience packages should be structured according to the taxonomies of product qualities, software engineering processes, and software engineering practices, thus facilitating easy retrieval of experience packages and continuous enlargement of the repository. There is no real tool support available but we advise use of a software-based archive called experience base. This experience base makes it possible to store, update, retrieve, and re-use experience within the organization applying the PROFES improvement methodology.

Certain aspects of **process modelling** are necessary in the PROFES improvement methodology. Examples of these are construction of the PPD models or description of the processes to facilitate assessment and improvement of an organization and its projects' software development processes. However, the PROFES improvement methodology does not prescribe or require the use of any specific process modelling technology. This is because there is no single process modelling approach that could meet all the needs of different users. Therefore, individual tools for process modelling may vary from simple textual description of the relevant processes to operation of extensive computer-supported modelling systems.

General Requirements for Supporting Tools

The following Table 6.1 summarizes the most data-intensive (sub)tasks for each main activity. These tasks define the most important functional requirements for supporting tools, and should therefore be addressed when establishing tool support for the PROFES methodology.

Table 6.1. PROFES phases and main activities with the most important functional requirements for tool support

PROFES phase	Main activity	Most data-intensive (sub)tasks
Characterize	<ul style="list-style-type: none"> Process assessment 	<ul style="list-style-type: none"> Management of assessment results and findings Generation of capability ratings, profiles, and assessment reports
	<ul style="list-style-type: none"> Process modelling 	<ul style="list-style-type: none"> Creation of descriptive process models
Set goals	<ul style="list-style-type: none"> PPD usage 	<ul style="list-style-type: none"> Utilization of existing PPD models Construction of new PPD models
	<ul style="list-style-type: none"> GQM planning 	<ul style="list-style-type: none"> Specification and management of both product improvement and GQM goals Definition and management of abstraction sheets Preliminary GQM plan management
Plan	<ul style="list-style-type: none"> Measurement planning 	<ul style="list-style-type: none"> GQM and measurement plans management
	<ul style="list-style-type: none"> Process modelling 	<ul style="list-style-type: none"> Creation of prescriptive process models
Execute	<ul style="list-style-type: none"> Data collection, analysis & presentation 	<ul style="list-style-type: none"> Collection and validation of measurement data Measurement data analysis Generation of measurement reports
	<ul style="list-style-type: none"> Process assessment and trend analysis 	<ul style="list-style-type: none"> Management of assessment results and findings Generation of capability ratings, profiles, and assessment reports Generation of trend analysis reports
Analyse	<ul style="list-style-type: none"> PPD model updating 	<ul style="list-style-type: none"> Management and validation of existing PPD models Construction of new PPD models
	<ul style="list-style-type: none"> Process modelling 	<ul style="list-style-type: none"> Management and validation of existing process models
Package	<ul style="list-style-type: none"> PPD packaging 	<ul style="list-style-type: none"> Construction of new PPD packages Management of existing PPD packages
	<ul style="list-style-type: none"> Experience packaging 	<ul style="list-style-type: none"> Construction of new experience packages Management of existing experiences

What about the actual tools then? Methodology development was the main objective of the PROFES project and so we did not develop any specific

PROFES toolset. Nevertheless, the tools introduced in this chapter were used and enhanced during the project. Each of these tools is explained more extensively in Appendix 5. Generally speaking, a tool selected to support a particular activity can range as required from traditional office packages to highly specific software tools. Specific tools should be considered if the amount of data to be collected for a particular activity is significant, or if data collection can be automated by using these tools.

Further Reading

See Appendix 5 for more detailed information on the tools used during the PROFES project. Here are some links to the organisations behind the tools:

Bootsample: <http://www.bootstrap-institute.com/>

GQMaspect: <http://www.iese.fhg.de>

MetriFlame: <http://www.ele.vt.fi/docs/soh/metriflame/>



Chapter 7

PRODUCT/PROCESS DEPENDENCIES IN PROFES

Product/Process Dependencies (PPDs) are the key element in PROFES that link product quality to the software process. A PPD describes how certain software processes, such as software design, are particularly important for achieving a high level of required product quality, for example reliability. When provided with a product quality goal and a collection of PPDs, project management can decide which processes should have the highest impact on the required product quality. Improvement actions, such as the introduction of new software engineering technologies, can then be focused on these processes. Using PPDs, improvement programmes can become particularly well controlled and effective.

This chapter introduces the idea of product/process dependency and describes the processes necessary to enable it to manage improvement programmes. The following topics are addressed:

- Introduction to PPDs
- PPD models
- PPD repositories
- PPD usage
- PPD development
- Development of PPD repositories
- Evolution of PPD repositories

Introduction to PPDs

A PPD is the relationship between aspects of a software process and the quality attributes of the product that is developed by this process. Information about PPDs is important for identifying improvement actions that are appropriate for attaining a given product quality goal.

Figure 7.1 shows an example PPD. It describes why the software design process is important for achieving high product reliability. When identifying improvement actions, it is useful to have additional information available on the process: (1) Which technology will help to achieve product quality? (2) In which contexts can the technology be effectively applied? For this reason, a PPD also involves a technology and a description of appropriate contexts. In this example, the technology is design inspections and the contexts are those projects with low or average overall time pressure.

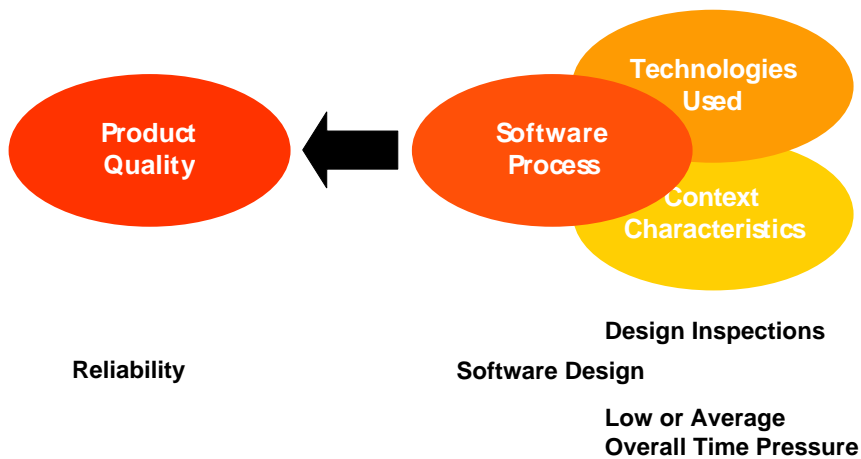


Figure 7.1. PPD example

PPD Models

A PPD model describes a product/process dependency in a structured and well-organized way, so that information can be effectively used for the identification of improvement actions. Table 7.1 shows an example of a PPD model. It consists of two basic parts: (1) A technology application goal that involves product quality, process, and technology. (2) A context model that describes contexts in which the impact of technology and processes on product quality will be effective.

The terms used for describing product quality, processes, and technology should be defined in a standard glossary. For instance, the ISO/IEC 9126 definitions of product quality attributes and the BOOTSTRAP 3.0 process

dimension are used as the baseline for product quality and process definitions in PROFES. The context model consists of attribute/value pairs that define contexts using the characteristics of individual context factors, such as overall time pressure and management commitment. A more detailed schema for defining PPD models is described below in the section on the development of PPD repositories.

Table 7.1. Example of a PPD model

PPD Model	
Product Quality	Reliability
Process	Software Architecture Design
Technology	Software Inspections
Context	
Overall time pressure in project	low average high
Experience of project team	low average high
Management commitment for inspections	low high

PPD Repositories

Software organizations that wish to accumulate and use their knowledge about product/process dependency should establish a corporate PPD repository. Such a PPD repository can simply be a paper document, such as a binder that is accessible to everyone who can benefit from PPD information. However, in order to ease access to the PPD repository, we recommend that it should be implemented in electronic form, e.g. as a database or web pages, and make it available through the corporate Intranet.

PROFES has developed such a PPD repository, based on the experience of the industrial application partners, literature, and other sources. A PPD repository is a key element in the PROFES improvement methodology. Figure 7.2 shows the role of a PPD repository within product-focused process improvement. A PPD repository documents experience from past software projects and offers this information for use during the planning of improvement programmes. New experience from these improvement programmes is fed back into the PPD repository for the benefit of future improvement programmes.

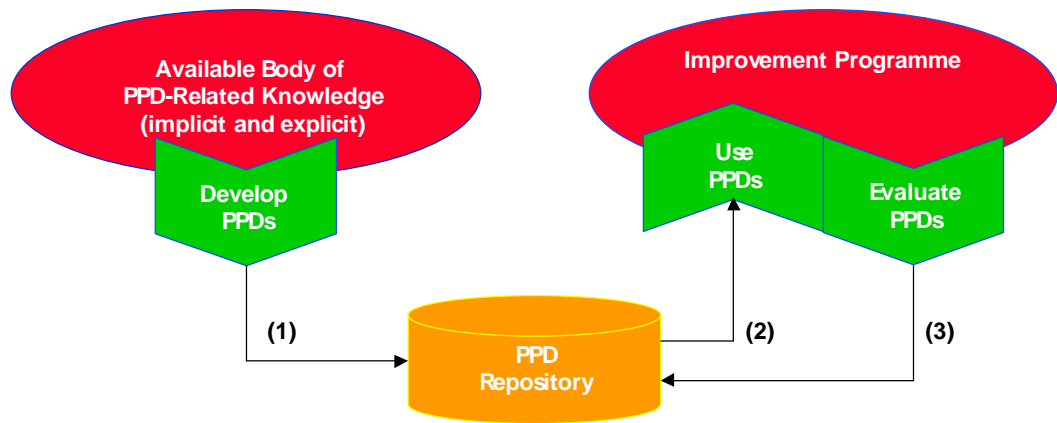


Figure 7.2. PPD lifecycle

PPD Usage

PPD repositories are mainly used within product-focused process improvement for selecting process changes to meet a given product quality goal. This is described in Step 6 of the PROFES improvement methodology in Chapter 3 of this manual.

In addition, PPDs can also be used for purposes such as:

- Focusing process assessments
- Management of technology-related project risks
- Organizing repositories of good software engineering practice.

In the following, each of these usage purposes of PPDs is briefly outlined.

Focusing Process Assessments

Process assessments may well require a considerable amount of effort. A PPD repository can be used for reducing the effort necessary for process assessments by identifying those processes that are particularly important for a given product quality goal. The assessment can then be limited to these selected processes. The following procedure describes how this selection can be performed. Its basic strategy is shown in Figure 7.3.

- Input: A product quality goal and a PPD repository
1. Retrieve all those PPD models from the PPD repository, whose product quality level is the same as the given product quality goal.

2. Review the context characteristics of these PPD models and check whether they match the project to be assessed.
 3. Remove those PPD models from the retrieved group that are not sufficiently well matched to the project to be assessed.
 4. For all remaining PPD models, identify and collect the referred processes. The collected list of processes contains all processes that are particularly important for attaining the given product quality goal within the given project context. These are the processes to be assessed during the focused process assessment.
 5. In some cases, it might be relevant to review the list of processes together with senior project members, in order to remove further processes or to add others that are not appropriately addressed in the current version of the PPD repository.
- Output: A list of processes to be assessed during the focused process assessment

Taken from the PROFES PPD repository, an example list of processes that are generally important for attaining high product reliability is shown in Table 7.2. For specific situations, the PROFES PPD repository suggests some additional processes. For instance, the process lifecycle methodology, system requirement analysis, and software requirements analysis are particularly important for attaining high reliability in projects that develop a new product line.

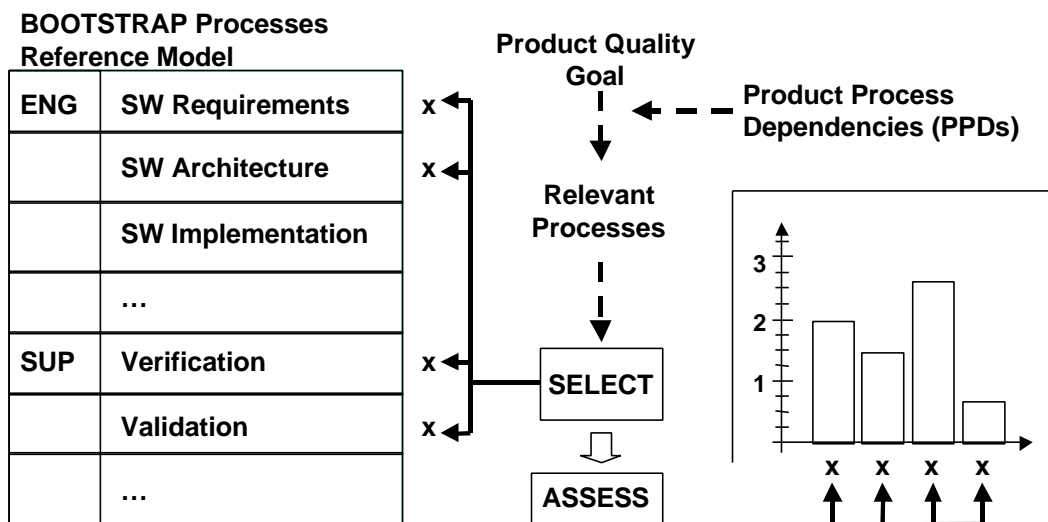


Figure 7.3. The strategy of focused assessments.

Table 7.2. Process-impact PPD models on “Reliability”

Process	Notes
Lifecycle Methodology	<ul style="list-style-type: none"> • PROFES-A: Iterative and incremental lifecycle methodology helps to ensure high reliability.
System Requirements Analysis	<ul style="list-style-type: none"> • PROFES-A: Requirements reviews, especially for new types of functionality, if the underlying hardware is new, or if the functionality is developed externally by a supplier
Software Requirements Analysis	<ul style="list-style-type: none"> • PROFES-A: Requirements reviews, especially for new types of functionality, if the underlying hardware is new, or if the functionality is developed externally by a supplier
System Architecture Design	<ul style="list-style-type: none"> • PROFES-A: Design reviews, especially for avoiding late changes and problems due to conflicting requirements • PROFES-C: Place particular emphasis on a thorough architectural design • SPICE: Relative importance wrt. Reliability can be concluded from the identified correlation of good design practices with the ability to satisfy specified requirements (the latter also includes reliability to some extent)
Software Architecture Design	<ul style="list-style-type: none"> • PROFES-A: Design reviews, especially for avoiding late changes and problems due to conflicting requirements • PROFES-C: Place particular emphasis on a thorough architectural design • SPICE: Relative importance wrt. Reliability can be concluded from the identified correlation of good design practices with the ability to satisfy specified requirements (the latter also includes reliability to some extent)
Software Detailed Design	<ul style="list-style-type: none"> • PROFES-A: Design reviews, especially for avoiding late changes and problems due to conflicting requirements, as well as assuring sufficiently detailed and correct specifications • TRG: Especially Cleanroom Software Engineering, Personal Software Process, and Software Inspections • SPICE: Relative importance wrt. Reliability can be concluded from the identified correlation of good design practices with the ability to satisfy specified requirements (the latter also includes reliability to some extent)
Software Implementation and Testing	<ul style="list-style-type: none"> • PROFES-A: Proper module test process and effective test tools • PROFES-C: Reuse existing and tested code modules • TRG: Especially Personal Software Process
Software Integration and Testing	<ul style="list-style-type: none"> • PROFES-A: Proper integration and system test processes, as well as effective test tools • TRG: Especially Cleanroom Software Engineering, and Personal

Process	Notes
	Software Process
System Integration and Testing	<ul style="list-style-type: none"> • PROFES-A: Proper integration and system test processes • TRG: Especially Cleanroom Software Engineering
Maintenance	<ul style="list-style-type: none"> • TRG: Especially Cleanroom Software Engineering, and Personal Software Process
Configuration Management	<ul style="list-style-type: none"> • PROFES-A: Assure consistent use of the correct work products and resolve change conflicts
Subcontractor Management	<ul style="list-style-type: none"> • PROFES-A: Select suppliers based on their domain experience, in order to achieve solid implementation of functionality, and place particular emphasis on the testing of the supplied modules
Reuse	<ul style="list-style-type: none"> • PROFES-C: Reuse existing and tested code modules

Technology-Related Risk Management

Risks associated with the introduction and application of software engineering technology is an increasingly frequent root cause of software project failure. However, new technology can also be a source of great success. Therefore, it is important to introduce new technology in a well-controlled manner that takes care of the identification and monitoring of technology-related project risks. PPDs are a suitable tool for such technology-related risk management.

The context models of PPDs contain the critical success factors for the successful application of software engineering technologies. Success is measured in terms of the achievement of certain product quality goals of interest. The explicit context characterizations of PPDs can be used as a checklist for identifying whether a given project fulfils the required success factors. The following procedure outlines how this can be accomplished.

- Input: A technology to be introduced, a process for which the technology is to be used, a product quality goal to be achieved, and a PPD repository
1. Retrieve those PPD models from the PPD repository that match the given technology, process, and product quality.
 2. Collect all context factors contained in the retrieved PPD models into a list, and develop a characterization questionnaire from this list.
 3. Characterize the project using the characterization questionnaire.

4. Identify deviations between the context characteristics of the retrieved PPD models and the actual project. These deviations indicate possible risks for technology failure.
 5. Develop a risk management plan for the project. The plan should contain risk mitigation or risk monitoring measures for each deviating context characteristic. Risk mitigation aims at making the project compliant with the context required by the technology. Risk monitoring aims at monitoring the relevant root causes of project risks and their effect when carrying out the project.
- Output: A risk management plan for technology-related project risks

An example comparison of a PPD model's context characteristics and a corresponding project characterization is shown in Table 7.3. The technology in this case is Software Inspections. The diagram indicates two deviations between project characteristics as required by the technology and the actual project characterization: (1) Degree of management commitment for inspections, and (2) overall time pressure during the project. The degree of management commitment can possibly be improved by talking to the relevant managers and making them aware of the importance of software inspection. This is an example of risk mitigation. Often, overall time pressure can not really be avoided. Therefore, it is important to monitor the project's adherence to schedule, and to regularly check whether software inspections are conducted according to plan. Such monitoring ensures that risks associated with the new technology are identified early on, and that corrective actions can be taken in time.

Table 7.3. Comparison of PPD and project characterization

Critical Success Factors of Software Inspections	Required Project Characteristics	Actual Project Characteristics	
Overall time pressure in project	low average high	low average high	!
Experience of project team	low average high	low average high	✓
Management commitment for inspections	low high	low high	!

Organizing Repositories of Good Practice

PPD models link software engineering technologies or practices with those processes and product qualities to which they can be applied. They also provide information about the contextual situations in which the technologies can be most effectively reused. This information is very suitable for organizing repositories of reusable software engineering technologies. It ensures effective knowledge retrieval from these repositories and supports

informed decision making during the planning of software projects and improvement programmes.

Hence, the basic structure of PPD repositories as described below in Appendix 3 can be used as a blueprint for organizing repositories of good software engineering practice. In some organizations, it might be more appropriate to establish such a repository of good practice before addressing the more complex issue of a full-fledged PPD repository. Gradually introducing the use of PPD repositories in this way can increase the buy-in for this new software engineering concept by the software development teams involved.

PPD Development

The development of PPD models is a combined analysis and design task. The main information sources are:

- Interviews with experienced software professionals
- Software measurement programmes
- Other project analysis methods and project documentation
- Literature
- Surveys

Usually, interviews are the basic information source, and can be combined with literature reviews, analysis of measurement data, and other techniques. In the following, a procedure for developing PPD models is outlined that integrates information from these sources:

1. Review literature sources for PPD information:
 - Textbook
 - Experience reports
2. Develop first tentative PPD models from this information.
3. Analyse project data available in the organization.
4. Refine and adapt the tentative PPD models based on the project data.
5. Acquire necessary additional information from experienced software engineering professionals in the target organization.

6. Systematically review the existing PPD models together with experienced software engineers from the target organization.
7. Adapt and modify the existing PPD models where necessary.
8. Possibly define new PPD models based on information from the review.

A paper form can be used for acquiring context characteristics of PPD models from experts, and is presented in Appendix 3.

PPD model development can also take advantage of existing PROFES PPD repositories. PPD models from this repository may be used as a starting point for the development of new PPD models adapted to a specific software organization. Therefore, the above procedure can be applied from Step 3 onwards.

Development of PPD Repositories

A PPD repository organizes a collection of PPDs, so that they can be deployed efficiently. Therefore, the development of PPD models should always be viewed in context with a PPD repository. There are four basic strategies for establishing a PPD repository. They differ with regard to what kind of related repositories (e.g. repositories of good software engineering practice) are available to an organization, or on which it wants to rely on:

- Build a PPD repository from scratch
- Adapt the PROFES PPD repository in order to obtain an initial, customized PPD repository
- Transfer an existing repository of good practice into a PPD repository
- Transfer an existing project repository into a PPD repository

The development of a PPD repository involves not only PPD model development, but also requires the definition of an appropriate index structure and a glossary of the terms used. These are important for providing efficient access to the PPD information stored in the repository. They are also a prerequisite for ensuring consistency between the possibly large number of individual PPD models in the repository. Information and assistance concerning such index structures and glossaries can be found in Appendix 3, which introduces the web-based PROFES PPD repository.

A pragmatic approach to the development of a customized, organization-specific PPD repository is to adapt an existing PPD repository, such as the PROFES PPD repository that can be accessed via the web¹. Adaptation can be carried out with the following procedure:

1. Obtain access to existing PPD models:
 - PROFES PPD repository
 - PPD repository from a different organizational unit in the company
2. Select the existing PPD models of interest.
3. Adapt terminology, product qualities definitions, product definitions, processes taxonomy, technology definitions, etc.
4. Systematically review the existing PPD models together with experienced software engineers from the target organization.
5. Adapt and modify the existing PPD models where necessary.
6. Possibly define new PPD models, based on information from the review.

The result of this procedure is a customized PPD repository that can gradually be refined and extended in order to reflect the progressive understanding of the organization's product/process dependencies.

Evolution of PPD Repositories

As the context of software development evolves, and new experience of the relationships between product quality and the software processes is gained, the PPD models need to be updated and refined in order to reflect this evolved context and experience. For this reason, we recommend that the application of software engineering technologies selected as PPD-based improvement actions be monitored. This monitoring will provide information on whether the improvement actions has been successful, i.e. whether the required product quality has been achieved, and whether the PPD models' context characterizations are consistent with the actual application context of the technology in the respective software project. Software measurement programmes (cf. further reading section) are an appropriate technique for PPD monitoring.

¹ www.iese.fhg.de/Profes

Project experience of PPD models can be reported using the technology application report template shown in Table 7.4. Such information provides a useful baseline for evolving a PPD repository. In addition, it provides narrative information about the application of software engineering technologies that can be useful for any future project planning and application of the technology.

There are three basic actions for evolving a PPD repository, depending on how well the respective technology has been applied in a project:

- No modification of the PPD repository / Update status of PPD models

No modification of the PPD repository is necessary, if either (1) the technology was applied well and the actual project context matched the PPD model's context characterization, or (2) the technology was not applied well and the actual context did not match the PPD model's context characterization.

In this situation, it might be useful to change a status attribute of the respective PPD model, in order to indicate that the PPD information has been confirmed through one or more software projects.

- Modify PPD models' context characterizations

The context characterizations of one or more PPD models in the repository could be updated in the following circumstances. Either (1) the technology was applied well, but the PPD model's context characterization did not match the actual project situation, or (2) the technology was not applied well, even though the PPD model's context characterization did match the actual project situation.

The update action can be the modification of an existing context characteristic, i.e. adding or deleting one or more occurrences of the context factor, or the addition or deletion of one or more context factor. For instance, if it turns out that the software engineers' experience is an important success factor for the effectiveness of a certain design method, then a respective context characteristic should be added to the PPD models of this technology.

- Add new technological variants and adapt context characterizations

In some cases, it might be necessary to slightly modify an already existing technology in a manner that can be defined as a variant of it. For instance, a variant of software inspections could be inspections without an inspection meeting.

Usually, the introduction of such a technology variant requires the modification of the initial technology's context characterizations, and the formulation of a new context characterization for the new technology variant. For instance, when it appears that software inspections are not very successful in geographically dispersed teams, and a variant of software inspections without inspection meetings is introduced, then all respective PPD models should be supplied with an additional context characteristic called *geographic distribution of project team*.

Updating a PPD repository in these three different ways will ensure that the repository always reflects the organization's actual knowledge of product/process dependencies within its software development projects.

Table 7.4. Template of a technology application report for providing PPD-related feedback for future projects

Technology Application Report	
Technology: <i>(Name of the technology)</i>	...
Project: ... <i>(Name and identifier of the project)</i>	
Product quality goal: ... <i>(What was the product quality goal of the project on which the technology was expected to have particular impact?)</i>	
Process: ... <i>(For which process has the technology been applied in order to contribute to the achievement of the above product quality attribute?)</i>	
Success of technology application: ... <i>(Did the technology application actually help in achieving the required product quality? If yes: Provide some evidence for the product quality achievement. If no: Why did the technology fail?)</i>	
Issues and difficulties encountered during technology application: ... <i>(Were some issues and difficulties encountered when applying the technology in the project? Explain briefly.)</i>	
Recommendations for future applications of the technology: ... <i>(What should be taken into consideration when applying the technology in future projects?)</i>	
Requests for updates of the PPD repository: ... <i>(Should information that refers to the technology in PPD models be changed?)</i>	
Author of this technology application report / Contact information: ... <i>(Name, Department, Telephone, E-Mail)</i>	
Date: ... <i>(MMM-DD-YY)</i>	

Further Reading

Adriana Bicego, Munish Khurana, and Pasi Kuvaja. BOOTSTRAP 3.0 – Software Process Assessment Methodology. In Proceedings of the SQM '98, 1998.

Andreas Birk, Pieter Derks, Dirk Hamann, Jorma Hirvensalo, Markku Oivo, Erik Rodenbach, Rini van Solingen, Jorma Taramaa: “Applications of Measurement in Product-Focused Process Improvement: A Comparative Industrial Case Study”. In Proceedings of the 5th International Symposium on Software Metrics (Metrics '98), IEEE Computer Society Press, Bethesda, Maryland, USA, November 1998.

Dirk Hamann, Janne Järvinen, Andreas Birk, and Dietmar Pfahl. A Product-Process Dependency Definition Method. In Proceedings of the 24th EUROMICRO Conference: Workshop on Software Process and Product Improvement, Vol. II, pages 898-904, IEEE Computer Society Press, 1998.

ISO/IEC. Information technology - Software product evaluation - Quality characteristics and guidelines for their use. ISO/IEC standard 9126. Geneva, Switzerland, 1991.



Chapter 8

BUILDING AN ORGANIZATIONAL INFRASTRUCTURE FOR PROFES

In order to apply the PROFES improvement methodology efficiently and effectively, a suitable infrastructure has to be in place in the organization. The design and implementation of such an organizational infrastructure is the topic of this chapter.

This purpose of this chapter is not to discuss **how** the infrastructure should be organized; rather the focus will be on **what** should be in place. Therefore this chapter will not discuss the technical infrastructure such as networks, computers, phones, etc., but will present the requirements for an organizational infrastructure in an improvement programme.

What is an organizational infrastructure?

An organizational infrastructure for PROFES is defined as *“the total set of human resources, skills, and organizational support that should be in place in order to carry out a PROFES improvement programme efficiently and effectively”*. Without such an organizational infrastructure, efforts spent on the improvement programme will be greater than necessary, results will be less than expected, and the risk of failure increases.

The objective of such an organizational infrastructure is to enable the proper application of the PROFES improvement methodology, and to facilitate its application so that the right effort is spent on the right tasks.

Experience Factory Concept

The background theory for the organizational infrastructure of PROFES is the Experience Factory, which defines an organizational concept for providing development projects with process and quality models. However, it focuses mainly on measurement projects.

This user manual is not limited to measurement, but expands the experience factory concept to include improvement programmes supported by the PROFES improvement methodology.

Basili et al. (1994) write: "*The Experience factory is a logical and/or physical organization that supports project developments by analysing and synthesizing all kinds of experience, acting as a repository for such experience, and supplying that experience to various projects on demand*". It packages experience by building informal or formal models and measures various software processes, products, and other forms of knowledge through people, documents, and automated support. The experience factory is illustrated in Figure 8.1.

The following roles are involved:

- Corporate management
- GQM definition team
- GQM analysis team
- Project team

Corporate management provides a project team with project objectives for software development, and a *GQM Definition team* with business goals. These business goals provide the long-term improvement targets to which the GQM team should work towards. The GQM Definition team translates these goals into measurement goals, and refines these measurement goals into metrics and measurement procedures, based on environmental characteristics provided by the project team and previous experience packaged in the *experience base*. The *project team* provides the *GQM Analysis team* with measurement data. The GQM Analysis team processes this data and presents the results to corporate management and the project team.

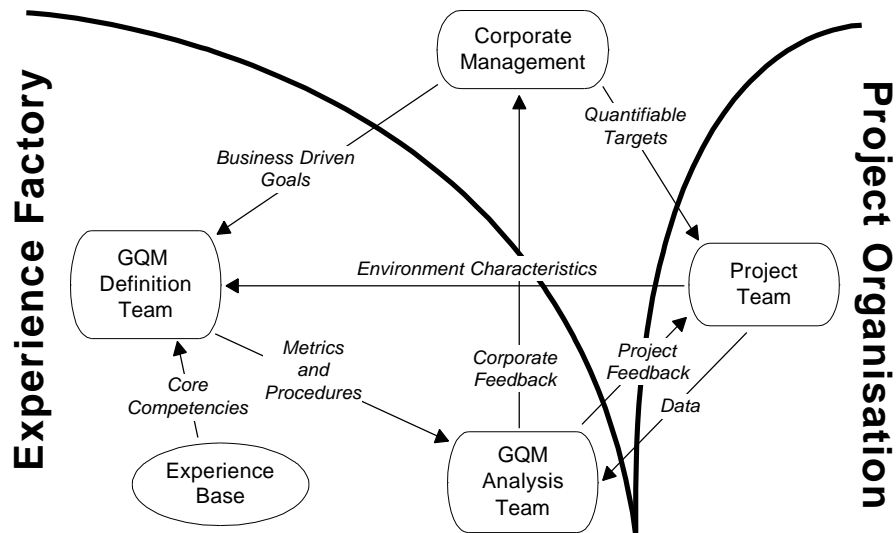


Figure 8.1. The Experience Factory (Basili et al, 1994)

The GQM Definition team maintains the information contained in the experience base, makes it efficiently retrievable, and controls and monitors access to it.

In this user manual, we make no distinction between the two teams as the experience factory concept does. The term 'PROFES team' is used for the independent organizational unit, which carries out those activities in an improvement programme that would not necessarily be carried out by the software development team (project team).

Setting Up the PROFES Infrastructure

This section presents those aspects that need to be covered by the PROFES infrastructure in order to support the PROFES improvement methodology.

Introduction

In the context of this user manual, the PROFES infrastructure is a system that supports process improvement operation in an organization that has decided to adopt the PROFES improvement methodology.

The PROFES infrastructure consists of the following elements:

- Dedicated human resources and competence
- Organizational support
- Reporting mechanisms

When the PROFES organizational infrastructure is put in place properly, the application of concepts, methods, techniques, and PROFES improvement methodology tools will be fully facilitated. Insufficient or unavailable support should not be the cause of any problems caused by insufficient or unavailable support will occur when applying the PROFES improvement methodology.

Although a proper infrastructure is necessary for the successful adoption of the PROFES improvement methodology, another important prerequisite is the total commitment of both management and the development project team. The project team (project manager and engineers) should be aware that the PROFES improvement methodology can only be effective with their full commitment and participation.

Requirements for a PROFES Organizational Infrastructure

This section contains some preferences for implementing the PROFES organizational infrastructure. However, it is not necessary to follow them entirely in every situation. Depending on the specific organization, different implementation strategies can be selected.

The requirements for an organizational infrastructure are:

- Several competencies are necessary to support the PROFES steps. Such competence includes process assessment, goal-driven metrics, process modelling, product assessment techniques, etc.
- Organizational support for managing organizational activities, including PPD experience base management. The PPD experience base is an organizational repository that should be managed across the project borders. PROFES also includes organizational activities, such as obtaining management commitment, understanding business goals, performing organizational process assessment, etc.
- Finally, persons neutral with the project should perform certain activities to avoid conflicts of interest. For instance, an external assessor should always carry out process assessment. This external assessor may belong to the organization but not the project team.

Establishing a PROFES Team

A separate PROFES team is established to support software development teams when product-driven process improvements are identified, implemented, and evaluated.

However, this separate team does not have to consist of dedicated personnel, as

- Certain activities need organizational support, i.e. managing the PPD repository
- Certain activities demand specific skills (process assessment, GQM), which can be provided either by a central organizational unit such as QA or SEPG, or also by external consultants
- Certain activities should remain independent of the project, such as the role of leading assessor, who can be from a central organization or be a consultant
- Certain activities must be internally managed by the organization, such as deploying management commitment, management reporting, and so on.

Small organizations that cannot afford a dedicated independent group can still apply the PROFES principles. In this case, all the above activities will be done by the people in the project teams themselves. To support this process, we would recommend that an external coach be appointed.

The PROFES team can operate in different ways, depending on the organizational structure, size, and culture. If possible, we recommend that the PROFES team remains independent of project targets, and so establishing a separate group has many advantages. Work is distributed between the PROFES team and the project team in such a way that time is only spent on improvement project tasks that critically depend on the project team's input or participation.

Such a PROFES team can be set up separately, but can also be integrated within an existing department, such as the software quality assurance department, or software engineering process group (SEPG). The PROFES team should:

- Remain independent of project teams, and have no interest in the improvement results
- Possess sufficient background knowledge of the products and processes being improved
- Bear in mind that as the most knowledgeable group on the project, the project team 'owns' the improvement programme
- Be improvement-driven, also with regard to self-improvement
- Be enthusiastic and motivate the project team
- Have a clear understanding of the business objectives.

The personal skills of people on a PROFES team play a crucial role in the success of the improvement programme, in particular their ability to motivate others. When a PROFES team lacks this ability, the whole improvement programme is at risk. A PROFES team should therefore include experienced staff, with sufficient background knowledge of software development and process improvement.

A PROFES team has three types of roles:

- 'Manager' – responsible for continuing the improvement programme and reporting to management
- 'Expert' – a person with full knowledge of the PROFES improvement methodology, who may be an external consultant
- 'Support' – supports the operational activities of the PROFES improvement programme.

An important requirement for successful improvement programmes is the level of mutual trust and co-operation between the PROFES team and the project team. Therefore, it is important that the PROFES team not be

dependent on the project team for process improvements and measurements. The objective of the PROFES team is to support improvement, rather than achieve it.

Furthermore, the PROFES team should regard itself as an facilitator of learning and be improvement-driven. It should respect the project team and allow it to carry out development tasks, as pre-defined procedures can not, nor will not always be followed. The PROFES team should retain an open mind on such issues, as in the end, only the developers are fully conversant with the products and processes, and are ultimately responsible for their project and its process, improvements, and measurements.

Allocate PROFES Resources

An improvement programme requires time. As such, it is an investment with the objective of raising product quality and/or process quality to a higher level. Resources need to be made available for such an investment. Without sufficient resources, it is very likely that not all the required tasks will be carried out, thus increasing the risk of failure.

The amount of resources required for a PROFES improvement programme will vary in different organizations. For a correct estimate, please refer to Chapter 5, PROFES Costs and Benefits, which is also included in this user manual. However, here are some rules of thumb to begin with:

- The engineers in the software development teams should make about 2% of their time available for the improvement programme. This effort is divided among the different tasks in the improvement programme.
- The PROFES team should have one full-time member available for every 30 engineers involved in the improvement programme on the software development teams. This number is only a very rough estimate, which can be adjusted to suit individual needs.

Define Management Reporting Process

In order to maintain management involvement and commitment, a management reporting process should be put into place. This process plans the frequency and content of the feedback to be provided. When such a process is defined up front, it becomes clear to both management and project teams in what way management will be informed of the improvement programme's progress. Such openness will makes matters much smoother, since no discussions need be held on this topic while the project progresses. Matters are defined, documented, and communicated to everybody.

It is possible to co-ordinate the management reporting process with other processes that already exist. For example, the management reporting process may be included in the ISO9000 management reviewing process, which organizes frequent managerial reviews. Improvement programme progress can then be presented for discussion during these meetings.

Organize Training and Promotion

Enthusiastic and motivated participants, who remain committed to its objectives, are crucial for the success of an improvement programme. To accomplish this, the PROFES team should organize regular training and promotion sessions during which:

- A clear definition of proposed improvement goals are presented
- Improvement benefits are explained
- The effect of the improvement programme on daily development activities is indicated
- Experience from other organizations/projects is discussed.
- The role of the project team is clearly explained.

If possible, all participants in the improvement programme should be present during these sessions. It is particularly important that those responsible for managing the project team are present, along with representatives of higher-level management.

An indication of the investment and its expected benefits should be given up-front for two reasons. Firstly, as the project manager needs to assign his people to the programme, he should plan their effort accordingly. Secondly, as the effort spent by the project team is considered to be an investment, it should not be undertaken if the expected revenues fail to cover the investment.

Once a project is approved, the project team members have to be trained for their respective roles in the project. To accomplish this, the PROFES team should organize training sessions during which

- PROFES improvement methodology principles are explained
- The PROFES improvement cycle is explained
- PROFES improvement steps are explained

The main focus of the training sessions is on explaining the PROFES improvement steps. The PROFES team should explain the particular

process steps to the project team members, and indicate to what extent they will be involved in these steps. After explaining the method, the improvement project plan should be presented to the project team members, indicating when their effort will be required. During these sessions, the PROFES team should not overemphasise the theoretical background of the PROFES improvement methodology, as it is more interested in hearing related practical issues and anticipates answers to questions such as:

- What improvement tasks should I perform?
- Why should I perform these tasks?
- How and when should I perform these tasks?
- How much effort will I have to make to perform these tasks?
- Will the tasks influence my daily activities?
- What do I get in return from this? What will I learn?

At least one training session should be organized during which the above points are discussed, and several sessions if necessary. All the people concerned in the project should be trained before project activities begin.

Further Reading

Victor R. Basili, Gianluigi Caldiera, and H. Dieter Rombach. "Experience Factory". In John J. Marciniak, editor, *Encyclopaedia of Software Engineering*, volume 1, pages 469–476. John Wiley & Sons, 1994.



Chapter 9

REFERENCES

Here you can find the most significant references to literature, and pointers for further reading on various topics relating to software process improvement.

- Adriaans, P., & Zantinge, D. 1996. *"Data Mining."* Addison-Wesley Publishing Company; ISBN: 0201403803.
- Althoff, K., Birk, A., Gresse von Wangenheim, C. & Tautz, C. *"Case-Based Reasoning for Experimental Software Engineering"*. In Lenz, M., Barsch-Spörl, B., Burkhard, H.-D., and Wess, S., editors: *"Case-Based Reasoning Technology – From Foundations to Applications"*, pages 235–254. Springer-Verlag, Berlin, 1998.
- Armitage, J. W. & Kellner, M. I. *"A Conceptual Schema for Process Definitions and Models"*. Proceedings of the 3rd International Conference on the Software Process (ICSP3), IEEE Computer Society, pp. 153-165, 1994.
- Bandinelli, S., Fuggetta, A., Lavazza, L., Loi, M. & Picco, G. P. *"Modeling and Improving an Industrial Software Process"*. IEEE Transactions on Software Engineering, Vol. 21, No. 5, pp. 440-453, May 1995.
- Basili, V. R., Caldiera, G. & Rombach, H. D. *"Experience Factory"*. In John J. Marciniak, editor, Encyclopaedia of Software Engineering, volume 1, pages 469–476. John Wiley & Sons, 1994.
- Basili, V. R., Caldiera, G. & Rombach, H. D. *"Goal Question Metric Paradigm"*. In John J. Marciniak, editor, Encyclopaedia of Software Engineering, volume 1, pages 528–532. John Wiley & Sons, 1994.
- Basili, V. R. & Caldiera, G. *"Improve Software Quality by Reusing Knowledge and Experience"*. Sloan Management Review, Fall 1995.
- Becker, U., Hamann, D. & Verlage, M. *"Descriptive Modeling of Software Processes"*. In *Proceedings of the Third Conference on Software Process Improvement (SPI '97)*, Barcelona, Spain, December 1997.
- Becker, U. & Webby, R. *"Towards a Comprehensive Schema Integrating Software Process Modeling and Software Measurement"*. Technical Report,

IESE021.97/E, Fraunhofer IESE, Kaiserslautern, July 1997.

- Bicego, A., Khurana, M., & Kuvaja, P. "*BOOTSTRAP 3.0 – Software Process Assessment Methodology*". In the Proceedings of the SQM '98, 1998.
- Birk, A., Giese, P., Kempkens, R., Rombach, D. & Ruhe, G. (Editors). "*The PERFECT Handbook (Vol. 1 - 4)*." Fraunhofer IESE Reports Nr. 059.97 - 062.97. Fraunhofer-Einrichtung für Experimentelles Software-Engineering, Kaiserslautern, 1997.
- Birk, A., Derks, P., Hamann, D., Hirvensalo, J., Oivo, M., Rodenbach, E., van Solingen, R. & Taramaa, J. "*Applications of Measurement in Product-Focused Process Improvement: A Comparative Industrial Case Study*". In Proceedings of the 5th International Symposium on Software Metrics (Metrics '98), IEEE Computer Society Press, Bethesda, Maryland, USA, November 1998.
- Birk, A., Kempkens, R., Rombach, H.D. & Ruhe, G. "*Systematic Improvement of Software Engineering Processes*", Proceedings of Fruehjahrstagung Wirtschaftsinformatik '98, Braunschweig: Vieweg, pp. 265-280, 1998.
- Birk, A. & Kiesgen, T. "*Identification of product/process dependencies from the SEI C4 technology reference guide*". Fraunhofer IESE Technical Report, Kaiserslautern, Germany, 1999.
- Birk, A. & Kröschel, F. "*A knowledge management lifecycle for experience packages on software engineering technologies*", in Proceedings of the Workshop on Learning Software Organizations, Kaiserslautern, Germany, June 1999.
- Birk, A. & Tautz, C. "*Knowledge management of software engineering lessons learnt*". Proceedings of the 10th International Conference on Software Engineering and Knowledge Engineering (SEKE), San Francisco, June 18-20, 1998.
- Briand, L., Differding, C. & Rombach, H. D. "*Practical guidelines for measurement-based process improvement*". Software Process Improvement & Practice, 2(4):253-280, December 1996.
- Burr, A. & Owen M. 1996. "*Statistical Methods for Software Quality: Using Metrics to Control Process and Product Quality* ", Coriolis Group (Sd); ISBN: 185032171X.
- Carey, P. & Berk, K. N. 1997. "*Data Analysis with Microsoft Excel*". International Thomson Publishing; ISBN: 0534529291
- Cios, K., Pedrycz, W. & Swiniarski, R. 1998. "*Data Mining Methods for Knowledge Discovery*." Kluwer, ISBN 0-7923-8252-8.
- Cody, R. P. & Cody, R. J. 1997. "*Smith Applied Statistics and the Sas Programming Language*". Prentice Hall College Div; ISBN: 0137436424
- Curtis, B., Kellner, M. & Over, J. "*Process Modelling*". Communications of the ACM, Vol. 35, No. 9, Sept. 1992.
- Dretzke, B. J. 1998. "*Statistics With Microsoft Excel*". Prentice Hall College Div; ISBN: 0139565337
- El Emam, K. & Birk, A. "*Validating the ISO/IEC 15504 measure of software requirements analysis process capability*". Fraunhofer IESE Technical Report IESE-

- 003.99. Kaiserslautern, Germany, 1999.
- Fayyad, U. M., Piatetsky-Shapiro, G., Smith, P. & Uthurusamy, R.. "Advances in Knowledge Discovery and Data Mining". MIT Press, Cambridge, Massachusetts, 1996.
- Greese, C., Hoisl, B., & Wüst, J. "A Process Model for Planning GQM-based Measurement". Technical Report, STTI-95-04-E, Software Technology Transfer Initiative (STTI), University of Kaiserslautern, October 1995.
- Hamann, D., Järvinen, J., Birk, A. & Pfahl, D. "A Product-Process Dependency Definition Method." In Proceedings of the 24th EUROMICRO Conference: Workshop on Software Process and Product Improvement, Vol. II, pages 898-904, IEEE Computer Society Press, 1998.
- Hardy, K. A. & Dilorio, F. C. 1996. "Quick Start to Data Analysis With Sas". Duxbury Pr; ISBN: 0534237606
- Hoffmann, M., Birk, A., van Els, A & Kempkens, R. "GQMAspect User Manual V1.0", Fraunhofer IESE, November 1996.
- Humphrey, W. "Managing the Software Process". SEI Series in Software Engineering. ISBN 0-201-18095-2, Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.
- Institute of Electrical and Electronics Engineers: "IEEE Standard for Reviews and Audits", IEEE Standard 1028-1988, 1989.
- ISO/IEC 9126: "Information technology – Software product evaluation – Quality characteristics and guidelines for their use". International Organisation for Standardisation (Ed.), Case Postale 56, CH-1211 Geneva, Switzerland, 1991. First edition 1991–12–15.
- ISO/IEC TR 15504-2: "Information Technology - Software Process Assessment - Part 2: A Reference Model for Processes and Process Capability". Technical Report type 2, International Organisation for Standardisation (Ed.), Case Postale 56, CH-1211 Geneva, Switzerland, 1998.
- Järvinen, J., Hamann, D. and van Solingen, R. "On Integrating Assessment and Measurement: Towards Continuous Assessment of Software Engineering Processes", in the Proceedings of METRICS'99, 1999.
- Kusters, R., v. Solingen, R. & Trienekens, J. "User-perceptions Of Embedded Software Quality", Chapter 4 (pp. 148-163) in: Software Quality from a Business Perspective - Directions and advanced approaches, Kluwer Bedrijfs Informatie, ISBN 90 267 2631 7, 1997.
- Kuvaja, P., Similä, J., Krzanik, L., Bicego, A., Saukkonen, S. & Koch, G. "Software Process Assessment & Improvement – The BOOTSTRAP Approach", Blackwell Publishers, 1994.
- van Latum, F., van Solingen, R., Oivo, M., Hoisl, B., Rombach, D. & Ruhe, G. "Adopting GQM-Based Measurement in an Industrial Environment". IEEE Software, 15(1), January 1998, pp. 78–86.
- MetriFlame, "Measurement and Feedback Tool of VTT Electronics". 1999– [online]. VTT Electronics, Finland. <http://www.ele.vtt.fi/docs/soh/metriflame/index.html>. [accessed 8.7.1999].

- Mitchell, T. 1997 "*Machine Learning*". McGraw Hill College Div; ISBN: 0070428077.
- Mollaghasemi, M. & Pet-Edwards, J. "*Making Multiple-Objective Decisions*". IEEE Computer Society Press, November 1998.
- Norusis, M. J. "*SPSS 8.0 guide to data analysis*." Upper Saddle River, NJ: Prentice-Hall, 1998. ISBN: 0-13-687484-3
- Parviainen, P., Järvinen, J. & Sandelin, T. "*Practical Experiences of Tool Support in a GQM-based Measurement Programme*", Software Quality Journal, Volume 6, No. 4, December 1997, pp. 238 - 294.
- Parviainen, P., Komi-Sirviö, S. & Sandelin, T. "*Measurement-Based Improvement of Critical Software Subprocesses: Experiences from two industrial cases*", European conference of Software Process Improvement, SPI '98, John Herriot, Monaco (1998), 10 p
- Paulk, M.C. et al. 1993. "*Capability Maturity Model for Software, Version 1.1*", Software Engineering Institute, Technical Report CMU/SEI-93-TR24.
- PROFES 1999, "*PROFES cost/benefit repository*." <http://www.iese.fhg.de/Profes>.
- PROFES 1999, "*PROFES PPD repository*." <http://www.iese.fhg.de/Profes>.
- Rombach, D. & Verlage, M. "*Directions in Software Process Research*". In Marvin Zelkowitz (ed.): *Advances in Computers*, Vol. 41, Academic Press, pp. 1-63, 1995.
- Sall, J., Lehman, A. & Saul, J. 1996. "*Jmp Start Statistics : A Guide to Statistical and Data Analysis Using Jmp and Jmp in Software*". Duxbury Pr; ISBN: 0534265650
- Schmidt's, J. 1998. Zinnote: "*Good Reports From Many Sources*." PC World magazine, May, 1998.
- Software Engineering Institute: "*C4 Software Technology Reference Guide – A Prototype*." Handbook CMU/SEI-97-HB-001, Software Engineering Institute, 1997.
- van Solingen, R. & Berghout, E. "*The goal/question/metric method, a practical method for quality improvement of software development*", McGraw-Hill ISBN 007-709553-7, 1999.
- SPICE Project. "*Phase 2: Trials*", interim report. 1998.
- Statsoft, "*Quick Statistica for Windows*". Paperback Windows 98 edition (January 1, 1999). Statsoft Inc; ISBN: 1884233120
- Zinnote, "*Data Integration and Reporting Toolkit*", Getting started. 1998 Positive Support Review Inc. <http://www.zinnote.com/Download/GettingStarted.pdf>



APPENDIX 1

AN OVERVIEW OF THE BACKGROUND ELEMENTS OF THE PROFES IMPROVEMENT METHODOLOGY

This appendix describes an overview of the five main background elements of the PROFES improvement methodology. These five background elements are fundamental to continuous learning and acquisition of experience from software projects and organizational units, and are the basis for systematic and continuous improvement. The first background element is the Quality Improvement Paradigm (QIP) [Basili et al. 1994a] that is the foundation for the phases of the PROFES improvement methodology. The following four background elements are also particularly important for systematic improvement: assessment of products and processes, goal-oriented measurement, process modelling, and Experience Factory.

- *Assessments* provide information on the current capability of the development processes and evaluate the products under development. When conducting an assessment, one is typically more interested in the broad and overall picture than in specific results of a detailed in-depth analysis. However, assessments can also be focused on or used together with measurements to monitor an improvement programme.
- *Goal-oriented measurement* measures products and processes to achieve the defined measurement goals. It provides the information necessary for understanding, assisting, and changing software processes in an organization. The Goal/Question/Metric (GQM) approach is a method for goal-oriented measurement that can be adapted to various improvement requirements and context characteristics. It ensures highly focused and efficient collection and use of measurement data, and builds on the involvement of the entire project team in measurement and improvement activities [Basili et al. 1994b], [Latum et al. 1998]. In addition to goal-oriented measurement, there are also several other

techniques for obtaining and analysing empirical information that can be relevant for improvement programmes (see for example [BT98]).

- Explicit *modelling of processes* is a prerequisite for identifying, storing, communicating, and utilizing experience within a software organization.
- *Experience Factory* addresses the storage, updating, retrieval, and adaptation of any relevant kind of experience and thus provides the basis for effectively making them available for software projects.

The Quality Improvement Paradigm (QIP)

The Quality Improvement Paradigm (QIP) is a comprehensive framework for systematically managing improvement by using experiments and measurement. QIP emphasizes that improvement programmes involve activities on two levels of the organizational hierarchy: The *strategic (organizational) level* and the *project level* (see Figure 1). Improvement on the strategic level is a continuous, organization-wide process that deals with long-term goals and issues that are relevant across the boundaries of a single project. Improvement on the project level is a short-term process that deals with the project-specific goals of improvement programmes. The activities on both levels should be closely integrated. This can be done by establishing two feedback cycles for software engineering information and experiences: the control cycle and the capitalization cycle [Basili & Caldiera 1995].

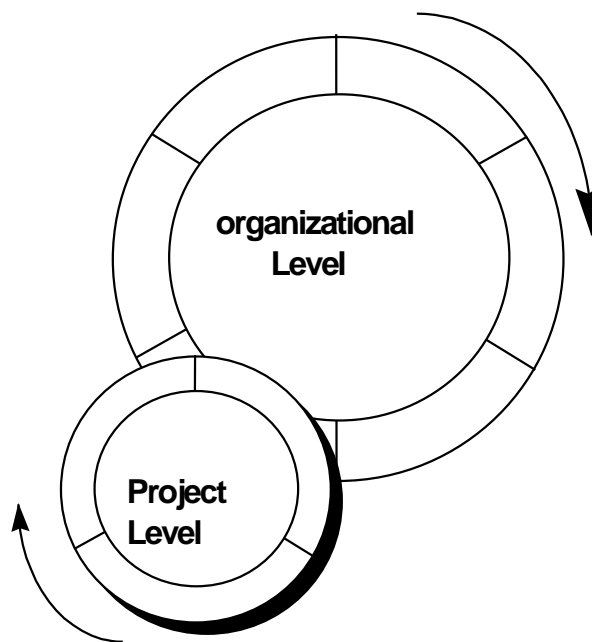


Figure 1. The organizational and project-level cycles of the Quality Improvement Paradigm.

The control cycle provides feedback to the project when the project is implemented. It provides analytic information about project performance at intermediate stages of development. The analytical information is deduced from empirical data collected during the course of a specifically designed measurement programme. The purpose of project-level feedback is to keep the project on track, to monitor the achievement of the project goals, and to indicate the necessity of corrective actions. The capitalization cycle provides feedback to the organization, i.e. across project boundaries. Its purpose is to understand what has taken place, and to accumulate

reusable experience in the form of artifacts, e.g. process definitions, software architectures, or cost models that can be reused by other projects.

The QIP helps to assist activities on the organizational (strategic) and project levels. For each level, a specialized version of the QIP has been defined. They are described in [Birk et al. 1997]. The project-level improvement process is integrated into the strategic-level improvement process, as achieving a long-term improvement goal usually involves several projects or pilot studies. Table 1 describes the project level and strategic level activities according to the six steps of the Quality Improvement Paradigm.

Table 1. Project level and strategic level activities within the QIP

QIP Steps	Project Level	Organizational (strategic) Level
Characterize	Describe type of project and identify relevant artifacts to be reused, i.e. techniques, methods, processes and models.	Describe type of organization and identify future trends.
Set Goals	Define project goals in measurable terms and derive related measures.	Define improvement goals in measurable terms.
Choose Models	Choose appropriate techniques, methods, processes and models, and develop the project plans including improvement activities.	Define improvement programme, i.e. identify (pilot) projects for investigating the hypotheses.
Execute Models	Perform project according to plan, collect data, and provide on-line feedback for project guidance	Perform (pilot) projects and collect data.
Analyse	Analyse project & collected data and suggest improvements.	Analyse (pilot) projects and evaluate achievement of improvement goals.
Package	Package analysis results into improved reusable artifacts.	Package experiences for use in future projects.

Product Assessment

Product assessment is needed a) to characterize the current product quality of an organization, and b) to compare these with the target product quality (see example in Figure 2), in order to define product improvement goals.

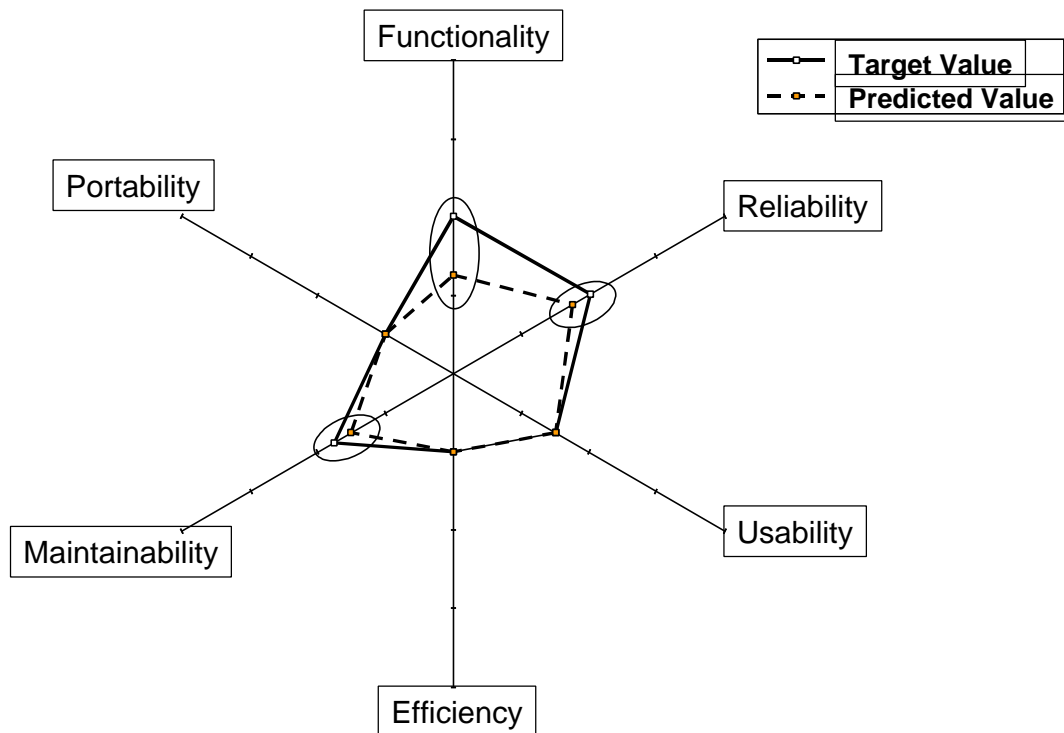


Figure 2. Example: Quality Needs vs Predicted Quality

Product quality has to be characterized in two steps:

- In the first step, a characterization structure is set up. This characterization structure can then be used to define target product quality. In the PROFES improvement methodology, the ISO 9126 standard is used as the kernel of a common characterization structure. In ISO 9126, software quality is classified according to the following quality characteristics: functionality, reliability, usability, efficiency, maintainability and portability. Including other factors related to product quality, such as development time and cost or time-to-market the kernel can be extended.
- In the second step, a measurement scale has to be chosen for each of the product quality characteristics of interest. This scale can then be used to define product quality quantitatively by assigning a value. A

measurement programme according to the GQM method can be used to define the measures and provide the measurement data.

There are several possibilities in assessing product quality. No specific method is proposed for the PROFES project, since goal-oriented measurement can always be used for assessing product quality. The MicroSCOPE method is more specifically adapted to perform a product quality assessment. MicroSCOPE is an enhanced version of the product assessment method developed in the ESPRIT project SCOPE. Checklists based on ISO 9126 quality characteristics are a key element of MicroSCOPE. The set of questions in the checklists provides a qualitative dimension, and the answers to them in the checklist provide a quantitative dimension.

The results of product assessment can serve as a starting point for the definition of the product (quality) improvement goals. The required or expected future product quality can either be defined internally by the SPU, or attempts can be made to derive the required target product quality target from the customer organizations. The SPACE-UFO method supports this type of customer-driven assessment of product quality requirements, and was developed in the ESPRIT project.

Process Assessment

Process assessment is used to characterize and evaluate the current development processes at the organizational and project levels, in order to find areas for process improvement.

In the PROFES project, the ISO 15504 (SPICE)-compliant assessment method BOOTSTRAP [Kuvaja et al. 1994] is used. The BOOTSTRAP methodology also supports the ISO 9001 standard and customer-focused software process assessment and improvement.

The BOOTSTRAP process assessment is performed both at organizational (SPU-Software Producing Unit) and project levels. Organizational-level assessment aims to assess the official written processes of the company. At project level, it assesses how these processes are carried out in practice. The strengths and weaknesses of current processes are identified in comparison to the ISO 15504 reference model underneath the assessment method.

The BOOTSTRAP software process assessment and improvement methodology was originally developed to improve the capability of European software-intensive industry. It was expected to meet ISO 9001 requirements when assessing small to medium-sized organizations. During the PROFES project, the BOOTSTRAP method was enhanced to meet the requirements set for embedded systems development assessment and improvement. The BOOTSTRAP assessment approach has been updated by introducing new process areas: product management, product (development) life cycle, and product-related processes.

Goal-Oriented Measurement

Measurement is a technique that supports the understanding, control, prediction, and improvement of software development processes and products. Goal-oriented measurement according to the Goal/Question/Metric (GQM) method [Basili et al. 1994b] represents a systematic approach to adapting and integrating the objectives of an organization into measurement goals and their refinement step-by-step into measurable values. GQM was chosen as an element of the PROFES improvement methodology, as it is today the most mature and widely used measurement approach (see [Briand et al. 1996], [Latum et al. 1998]). By applying GQM, information is identified that is relevant for solving specific problems (goals), and that can be represented in a practical, applicable, and interpretable way. With GQM, the measurements focus only on those pieces of information that were previously derived from an explicitly described problem statement. The problem statement (or goal) and its associated measurement model are formally defined and described in the GQM plan. A GQM plan contains three levels of abstraction:

- Conceptual level or goal

A goal is defined for an object such as a process or product. This goal indicates the measurement purpose, e.g. characterization, prediction, or control, the properties of the object that are of interest (the quality focus), the viewpoint, i.e. the roles for which the measurement is made, and the environment in which the measurement data is collected, analysed, and interpreted.

- Operational level (question)

A set of questions is used to achieve the goal on an operational level, and to specify the way in which measurement data will be interpreted.

- Quantitative level (metrics, or preferably *measures* in this document)

A set of measures is associated with every question in order to answer it in a quantitative way. A measure may contribute to different questions, and one question may be typically answered by taking into account several measures. Measurement data may result from objective or subjective measurement, and measurement can be done on different types of scales (nominal, ordinal, interval, etc.).

A GQM model, as defined in the GQM plan, is a hierarchical structure beginning with a goal. The goal is refined into several questions that usually divide the issue into its major components (cf. Figure 3). Each

question is then refined into measures, some of them objective and some of them subjective measures of interest.

The GQM planning process is divided into four stages [Gresse et al. 1995]. The first stage is to identify a set of measurement quality goals. The second stage derives questions that define the goals as completely as possible. The next stage consists of specifying the measures that need to be collected in order to answer those questions, and to track the conformance of products and processes to the goals. After the GQM model has been specified, it is necessary to develop mechanisms that collect measurement data. This is described in the measurement plan and the associated data collection forms. Tool support is available for the development of the GQM plan, data collection, storage, and visualization (MetriFlame), as well as data analysis (standard statistics packages like Statistica, SAS, etc.).

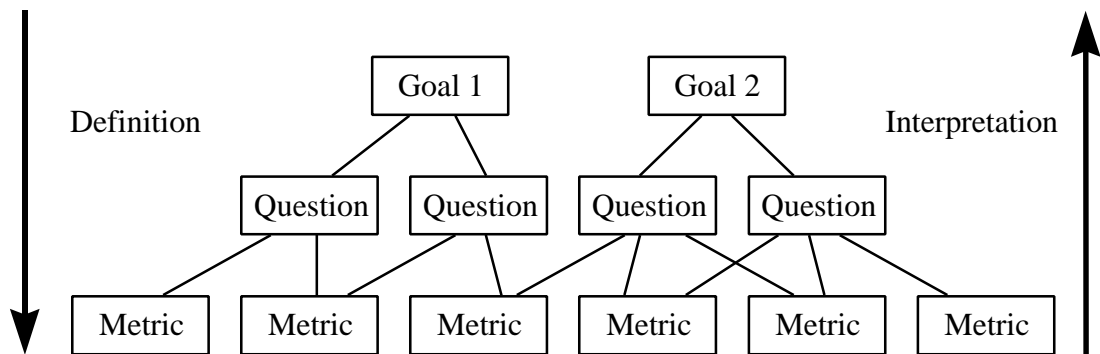


Figure 3. The hierarchical structure of a GQM model.

Process modelling

There are two types of process modelling activities [Curtis et al. 1992]: descriptive and prescriptive. The task of prescriptive process modelling is to design a new process or to define an intended process improvement for the purpose of assistance. The conformance of process implementation to the prescriptive process model may be enforced through the support of a tool-based Software Engineering Environment (SEE). The task of descriptive process modelling is to capture the current software development practices and organizational issues of a software-producing unit as a Descriptive Process Model [Becker et al. 1997]. The purpose of descriptive process modelling is to support human communication and to assist analysis of the current process.

The content of a descriptive process model is mainly based on collecting knowledge from process experts and software development practitioners, i.e. the actual processes are represented, not the official ones [Bandinelli et al. 1995]. Entities and relationships between entities, e.g. input/output relations, activity sequences, reporting channels between roles, and role assignments to activities represent relevant real-world aspects. Entities are formalized in an operational way through attributes, which characterize the entities. Examples of attributes are size, complexity, status, time, and effort.

Different approaches to descriptive process modelling can be applied. Since there are many forms of process modelling, no particular recommendation for a specific method will be given in the PROFES improvement methodology. Only a set of the most important types of entities that are typically contained in a descriptive process model are listed, based on the conceptual framework suggested in [Armitage & Kellner 1994]:

- Artifacts consumed and produced
- Activities carried out
- Agents (with roles) involved
- Tools used
- Technologies, techniques, and methods used
- Relationships between activities and artifacts, i.e. flow of artifacts
- Assignment of roles to activities
- Usage of tools in activities
- Application of technologies/techniques/methods in activities
- Relationships between products, i.e. product hierarchies
- Relationships between roles, i.e. communication network.

A comprehensive formal schema for process models that supports integration with measurement was recently defined in [Becker & Webby 1997].

The PROFES improvement methodology does not suggest specific techniques or a method for process modelling. This is due to the fact that there have been many different techniques and methods suggested in literature ([Curtis et al. 1992] provides an overview), and most companies already use one or another process modelling method. Process modelling methods widely used in practice are mainly based on simple textual or graphical forms of representation, such as ordinary natural language, structured natural language, template-oriented textual descriptions, flowcharts, activity diagrams, data flow diagrams, and SADT diagrams. More advanced techniques and methods that also provide tool support specifically adapted for knowledge elicitation, model representation, and process simulation, include Petri-Net-based approaches using Process Weaver, and state chart based approaches using Statemate.

Experience Factory

The organizational model of the PROFES improvement methodology facilitates comprehensive reuse of artifacts and models, and is a refined and extended version of the Experience Factory organization [Basili et al. 1994a].

The Experience Factory is an organizational learning infrastructure for software development. Its main part is the experience base, a corporate repository for storing relevant software engineering experience. It distinguishes the project organization in which the software development activities are performed, from the organizational improvement infrastructure, where analysis and packaging activities are carried out for project support and for maintenance and continuous evolution of the experience base (cf. Figure 4).

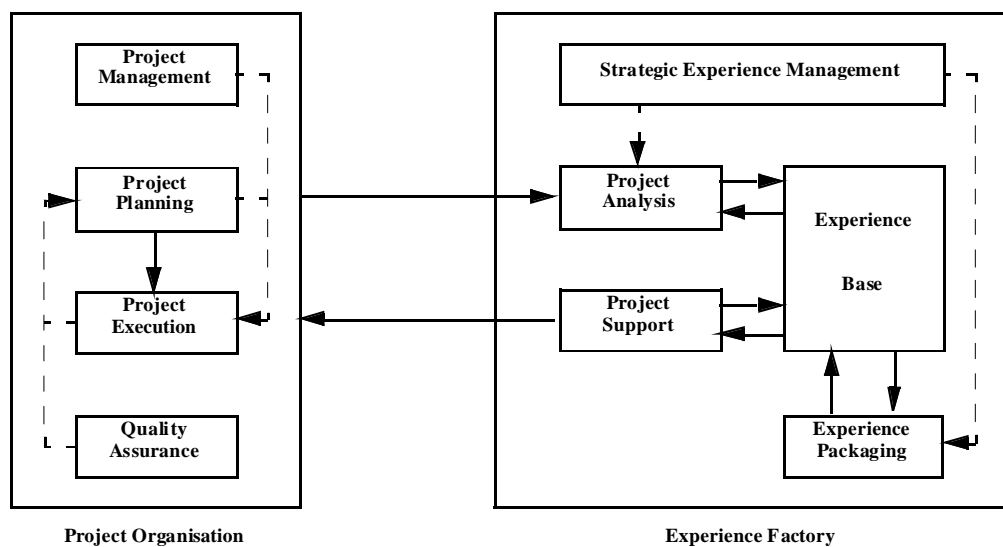


Figure 4. The Experience Factory organization (cf. [Basili et al. 1994a]).

References to Appendix 1

- [Armitage & Kellner 1994] J. W. Armitage and M. I. Kellner: "*A Conceptual Schema for Process Definitions and Models*". Proceedings of the 3rd International Conference on the Software Process (ICSP3), IEEE Computer Society, pp. 153-165, 1994.
- [Bandinelli et al. 1995] S. Bandinelli, A. Fuggetta, L. Lavazza, M. Loi, and G. P. Picco: "*Modeling and Improving an Industrial Software Process*". IEEE Transactions on Software Engineering, Vol. 21, No. 5, pp. 440-453, May 1995.
- [Basili et al. 1994a] Victor R. Basili, Gianluigi Caldiera, and H. Dieter Rombach. "*Experience Factory*". In John J. Marciniak, editor, Encyclopaedia of Software Engineering, volume 1, pages 469-476. John Wiley & Sons, 1994.
- [Basili et al. 1994b] Victor R. Basili, Gianluigi Caldiera, and H. Dieter Rombach. "*Goal Question Metric Paradigm*". In John J. Marciniak, editor, Encyclopaedia of Software Engineering, volume 1, pages 528-532. John Wiley & Sons, 1994.
- [Basili & Caldiera 1995] V.R. Basili, G. Caldiera. "*Improve Software Quality by Reusing Knowledge and Experience*". Sloan Management Review, Fall 1995.
- [Becker et al. 1997] Ulrike Becker, Dirk Hamann, and Martin Verlage. "*Descriptive Modeling of Software Processes*". In *Proceedings of the Third Conference on Software Process Improvement (SPI '97)*, Barcelona, Spain, December 1997.
- [Becker & Webby 1997] Ulrike Becker and Richard Webby: "*Towards a Comprehensive Schema Integrating Software Process Modeling and Software Measurement*". Technical Report, IESE021.97/E, Fraunhofer IESE, Kaiserslautern, July 1997.
- [Birk et al. 1997] A. Birk, P. Giese, R. Kempkens, D. Rombach, G. Ruhe (Editors). "*The PERFECT Handbook (Vol. 1 - 4)*." Fraunhofer IESE Reports Nr. 059.97 - 062.97. Fraunhofer-Einrichtung für Experimentelles Software-Engineering, Kaiserslautern, 1997.
- [Birk and Tautz 1998] Andreas Birk and Carsten Tautz: "*Knowledge management of software engineering lessons learnt*". Proceedings of the 10th International Conference on Software Engineering and Knowledge Engineering (SEKE), San Francisco, June 18-20, 1998.
- [Briand et al. 1996] Lionel Briand, Christiane Differding, and H. Dieter Rombach: "*Practical guidelines for measurement-based process improvement*". Software Process Improvement & Practice, 2(4):253-280, December 1996.
- [Curtis et al. 1992] B. Curtis, M. Kellner, J. Over: "*Process Modelling*". Communications of the ACM, Vol. 35, No. 9, Sept. 1992.
- [Gresse et al. 1995] Christiane Gresse, Barbara Hoisl, and Jürgen Wüst: "*A Process*

Model for Planning GQM-based Measurement". Technical Report, STTI-95-04-E, Software Technology Transfer Initiative (STTI), University of Kaiserslautern, October 1995.

- [Kuvaja et al. 1994] P. Kuvaja, J. Similä, L. Krzanik, A. Bicego, S. Saukkonen, G. Koch. "*Software Process Assessment & Improvement – The BOOTSTRAP Approach*", Blackwell Publishers, 1994.
- [Latum et al. 1998] Frank van Latum, Rini van Solingen, Markku Oivo, Barbara Hoisl, Dieter Rombach, and Günther Ruhe. "*Adopting GQM-Based Measurement in an Industrial Environment*". IEEE Software, 15(1), January 1998, pp. 78–86.



APPENDIX 2

THE PROFES TEMPLATES

Introduction

When an organization uses an improvement methodology, it will need to prepare documents such as plans, process and product models, and reports. The PROFES improvement methodology provides detailed guidelines to support those organizations willing to apply it. The guidelines include an indication of the output that will result from execution of the PROFES phases and steps and recommend documents where such output can be collected. These documents include assessment and GQM plans, specifications of product quality requirements and goals, process and product assessment reports, feedback session reports, and experience packages. To assist and apply the PROFES improvement methodology more efficiently in practice, this section provides a set of templates for the main documents produced during a PROFES improvement cycle. A template is a skeleton of a document with guidelines on how to complete each section and, whenever possible, also with examples of text that can be directly used to complete the document.

Several organizations already have documentation in place as part of their quality systems and improvement initiatives. The purpose of this section is not to recommend an alternative set of documents, but to provide skeletons that each organization can adapt to its own needs and can be integrated with already existing plans and reports. The PROFES improvement methodology itself is a tailorable methodology, as are its templates. Adjusting the PROFES templates to specific organizational needs is one step of the tailoring process that we recommend to an organization, in order to get the best of PROFES while retaining previously adopted and successful procedures and methods.

Overview of the PROFES Templates

The following is a list of the templates provided in the following sections to support the application of the PROFES improvement methodology:

- Appendix 2.1. Product quality needs report**
- Appendix 2.2. Process assessment plan**
- Appendix 2.3. Process assessment report**
- Appendix 2.4. Process assessment effort collection template**
- Appendix 2.5. PPD model template**
- Appendix 2.6. Process improvement plan**
- Appendix 2.7. GQM plan, including a template for defining GQM goals and abstraction sheets**
- Appendix 2.8. Measurement plan**
- Appendix 2.9. GQM effort collection template**
- Appendix 2.10. Feedback session report**
- Appendix 2.11. Technology application report**

Table 1 shows the PROFES templates and the PROFES phases and steps during which they are generated. In many cases, one individual document is prepared for several PROFES steps and sometimes several PROFES phases.

Table 1. The PROFES Phases, Steps, and Templates

PROFES phases	PROFES steps	Templates
<i>Characterize</i>	1. Verify commitment	
	2. Identify product quality needs	Product quality needs report
	3. Determine current product quality	Product quality needs report
	4. Determine current process capability	Process assessment plan, process assessment report, process assessment effort collection template
<i>Set Goals</i>	5. Set product improvement goals	PPD model template, process improvement plan
	6. Determine necessary process changes	Process improvement plan
<i>Plan</i>	7. Describe process changes	
	8. Set metrics for the processes and product	GQM plan (incl. GQM goal definition and abstraction sheet), measurement plan, and GQM effort collection template.
	9. Prepare improvement implementation	Process improvement plan
<i>Execute</i>	10. Implement and monitor improvements	Feedback session report and GQM effort collection template
	11. Evaluate results	Technology application report and GQM effort collection template
<i>Package</i>	12. Update experience base	GQM effort collection template

Notes:

- The product quality needs report, as it is presented here, is a comprehensive document that includes specification process output for identifying product quality requirements, as well as the results of a product quality assessment. The use of this document depends greatly on the organizational structure and characteristics of the development project. It may be necessary to separate it into two main parts if the product quality requirements and the current product quality are surveyed in detail at this stage. The investigation of product quality needs and goals may also be part of an initial feasibility study or user requirement specification effort. On the other hand, the product assessment survey may simply be a synthesis of quality control activities performed systematically during the development cycle
- The process assessment plan is used to plan assessment and GQM interviews
- The process assessment report provides guidelines for completing a SPICE-conformant process assessment report
- The process assessment effort collection template supports collection of effort data for a BOOTSTRAP process assessment
- The PPD model template is used to elicit PPD knowledge from engineers and/or to document hypothetical and validated PPDs
- The process improvement plan is a central document that summarizes the results of several activities: it contains a detailed description of process improvement goals, and plans to implement and validate them. It is a major reference for improvement monitoring and validation
- The GQM plan contains a complete specification of the measurement goals according to the GQM method. The template includes guidelines for specifying a GQM goal, completing abstraction sheets and preparing a GQM plan for each identified measurement goal
- The measurement plan specifies detailed procedures for collecting measures according to the corresponding GQM plans
- The GQM effort collection template supports the collection of effort data for a GQM measurement programme, including activities from planning to packaging
- The feedback session report contains results, conclusions, decisions, and action items from each feedback session performed during the improvement project

- The technology application report is recommended for reporting experience from application of a specific technology in a project or improvement programme. Technology application reports complement the PPD repository.

References to Appendix 2

Multi-party chain techniques:

R. Kusters, R. v. Solingen, J. Trienekens, *“User-perceptions Of Embedded Software Quality”*, Chapter 4 (pp. 148-163) in: *Software Quality from a Business Perspective - Directions and advanced approaches*, Kluwer Bedrijfs Informatie, ISBN 90 267 2631 7, 1997.

More information on MPC can be downloaded from the Internet page: ‘<http://www.tm.tue.nl/vakgr/it/mwerkers/rso/rini.htm>’. SPICE-conformant assessment:

ISO/IEC TR 15504-2: *“Information Technology - Software Process Assessment - Part 2: A Reference Model for Processes and Process Capability”*. Technical Report type 2, International Organisation for Standardisation (Ed.), Case Postale 56, CH-1211 Geneva, Switzerland, 1998.

Guidelines for Using the PROFES Templates

The rest of Appendix 2 contains the PROFES templates.

Each template has the following structure:

- A proposed table of contents is provided at the beginning of each template
- Suggestions for completing each section are then provided. Suggestions may include:
 - guidelines for completing the section
 - predefined text that can be directly included in the final document
 - predefined text that needs to be completed before it can be included in the final document
 - examples

In order to differentiate clearly between guidelines, predefined text, and examples, the following formats have been used:

- Guidelines are included in a box, introduced by the word “Guideline” and use normal text
- Predefined text is provided as normal text
- Predefined text to be completed uses normal text. Those parts to be completed are included in <> brackets
- Examples are written in italics



APPENDIX 2.1

THE PROFES TEMPLATE FOR: PRODUCT QUALITY NEEDS REPORT

Table of Contents

1. Introduction
2. Product quality needs Investigation
 - 2.1 Identification of Relevant Users
 - 2.2 Detailed Descriptions of Quality Requirements
3. Evaluation of the Current Product Quality
4. Identification of the Target Product Quality
5. Expected Product Quality
6. Wanted towards Current, Target and Expected Quality Profile
7. Product Improvement Goals

1. Introduction

Guideline

Product quality goals and product improvement objectives are generally defined as a result of a process that involves several roles outside and inside the organisation. This process may be carried out outside the boundaries of the organisational unit applying the PROFES improvement methodology. In such cases product quality goals may be provided as input requirements for development projects. In other cases the full specification process may be in the scope of the organisational unit applying the PROFES improvement methodology. This template provides synthetic guidelines to ensure that product quality needs are clearly identified.

2. Product quality needs Investigation

2.1 Identification of Relevant Users

Guideline

Relevant *users* that can provide quality requirements for the final product are identified. Relevant users are not only the final users, i.e. those that will either operate the product or use it directly, but also organisational entities like the company management, marketing, quality assurance, manufacturing department and external organisations like sector specific certification bodies. One method to identify the relevant users is using a Multi-Party Chain chart.

2.2 Detailed Descriptions of Quality Requirements

Guideline

Based on interviews of the relevant users detailed user quality requirements should be identified. Requirements should be expressed in a quantitative way, using a well-identified metric and expressing the target value. Mapping of requirements on extended ISO 9126 characteristics and sub-characteristics (the standard ISO 9126 characteristics plus cost-effectiveness, timeliness and time-to-market) should be provided. Detailed requirements can be represented in a table format as follows:

<User name 1>				
Requirement	(Extended) ISO 9126 characteristics	ISO 9126 sub-characteristic	Metric	Target value
...
...
<User name 2>				
Requirement	(Extended) ISO 9126 characteristics	ISO 9126 sub-characteristic	Metric	Target value
...

3. Evaluation of the Current Product Quality

Guideline

When a new version or release of an existing product is being developed, the quality of the final product will largely depend on the quality of the current version of the product. An assessment of the current product quality should be performed using ISO 9126 characteristics and sub-characteristics. A GQM plan can be developed to define which metrics should be used to evaluate relevant quality characteristics and sub-characteristics.

4. Identification of the Target Product Quality

Guideline

Based on the “wanted” product quality as derived from the user requirements, the current product quality and available time and resource constraints a “target” product quality profile will be defined. The target product quality profile will set the quantitative objectives for the development project. Product quality objectives should be expressed using ISO 9126 characteristics and sub-characteristics. A GQM plan can be

developed to define metrics to evaluate relevant quality characteristic and related target values.

5. Expected Product Quality Profile

Guideline

The final product quality will largely depend on the processes applied to develop it. A profile of *expected product quality* could be derived in a qualitative way based on the process assessment results.

Additionally the expected product quality can be derived from an analysis of previous measurement data showing the baseline product quality level for a defined context (for instance in the same department for similar projects).

6. Wanted towards Current, Target and Expected Quality Profile

Guideline

A comparison of the *wanted*, *current*, *target* and *expected* product quality profile is useful to show weak areas where actions are needed if goals are to be achieved.

The comparison may be done using bar charts or Kiviat diagrams.

7. Product Improvement Goals

Guideline

In this section product improvement goals are identified with reference to ISO 9126 characteristics and sub-characteristics. A clear and complete specification of each improvement goal should be provided.



APPENDIX 2.2

THE PROFES TEMPLATE FOR: ASSESSMENT PLAN

Table of Contents

1. Introduction
2. Assessment Purpose
3. Assessment Scope
 - 3.1 Involved Organisational Units
 - 3.2 Processes to Assess
 - 3.3 Key Personnel to interview
 - 3.4 Reference documentation
4. Assessment Constraints
5. Responsibilities
 - 5.1 Sponsor
 - 5.2 Facilitator
 - 5.3 Assessment Team

5.4 GQM coach

6. Interview Plan and Schedule

1. Introduction

Guideline

This section specifies the organisational framework and context within which the process assessment will be performed. If GQM interviews will be performed together with assessment interviews this should be mentioned.

2. Assessment Purpose

Guideline

The main objectives and specifically improvement objectives of the organisation and involved projects are listed in this section. The main improvement objectives for the final product will be specified with reference to quality characteristics and sub-characteristics, time and cost constraints. Objectives will be specified for the overall organisation and or specified projects as needed. Objectives specified in this section will drive the selection of processes to assess.

3. Assessment Scope

3.1 Involved Organisational Units

Guideline

This section specifies which organisational units will be addressed in the process assessment. Organisational units can include departments, business units, functional or product areas and projects.

3.2 Processes to Assess

Guideline

This section will specify which processes will be assessed. The selection of processes is driven by the objectives identified in the purpose section. Processes will be selected with reference to the process model of the process assessment methodology adopted and will be mapped onto the

real processes of the target organisational unit. All processes in the process model can be selected (full process assessment) or a subset of processes can be identified (focused process assessment). A complete process assessment is recommended when an overall understanding of process capability is needed, i.e. when a process assessment is performed either for the first time or a long time after the previous one. When a focused assessment is performed, selection of the relevant processes to assess is facilitated by use of the PPD repository. PPD models will guide the identification of processes that have proved to affect the organisational unit objectives.

Product-process relationships that have driven the selection of processes to assess can be summarised in a table as follows:

Goal	Ranking	Related process
<ISO 9126 characteristic>	<2>	<Process 1>
	<1>	...
	<3>	<Process n>
Cost effectiveness	<1>	<Process x>
...		
Timeliness	<1>	<Process z>

In the above table goals can be picked from ISO 9126 characteristics and sub-characteristics plus timeliness, time-to-market and costs-effectiveness. Processes are selected from the process model of the selected process assessment methodology. Ranking shows the expected level of impact of each process on the relevant goal. The ranking depends on contextual factors.

3.3 Key Personnel to interview

Guideline

Key persons to interview for each process both at organisational and project level will be specified in this section.

3.4 Reference documentation

Guideline

Reference documentation to support the process assessment will be specified, including existing organisational procedures and work instructions, methodologies, project plans and deliverables.

4. Assessment Constraints

Guideline

This section will specify any constraint that may affect the process assessment execution, including possible unavailability of key personnel, time constraints, and limitations in the available information.

5. Responsibilities

5.1 Sponsor

Guideline

The sponsor is the organisational unit and or role in the organisation that commits the process assessment execution. The sponsor is the owner of process assessment results and is responsible for ensuring that all necessary information and personnel will be available so that the process assessment can be completed successfully. Provide the name and affiliation of the sponsor.

5.2 Facilitator

Guideline

The facilitator plays the role of interface between the assessment team and the organisational unit involved in the process assessment. Therefore the facilitator is normally a person from the same organisational unit. His role is to facilitate assessors in effectively access all information they need to take into consideration and all persons that can contribute to better understand the nature and capability of assessed processes.

5.3 Assessment Team

Guideline

Names, affiliation and qualification of assessors will be included in this section. The assessment team normally includes a lead assessor, qualified against the qualification schema of the selected process assessment methodology, and an assessor, who must be at least trained and experienced in the selected assessment methodology. Both persons playing the role respectively of lead assessor and assessor are part of the PROFES team.

5.4 GQM Coach

Guideline

GQM interviews can be conducted together with assessment interviews. This section should specify who will play the role of GQM coach. The person playing the role of GQM coach is part of the PROFES team.

6. Interview Plan and Schedule

Guideline

Objectives and main topics for each interview are specified, taking into account the need of process assessment and the needs of GQM interviews.

A schedule of interviews is provided.



APPENDIX 2.3

THE PROFES TEMPLATE FOR: PROCESS ASSESSMENT REPORT

Table of Contents

1. Introduction
2. Assessment Context
 - 2.1 Purpose
 - 2.2 Validity
 - 2.3 Assessment Scope
 - 2.4 Assessment Constraints
 - 2.5 Responsibilities
3. Assessment Results
4. Strengths and Weaknesses
5. Improvement Recommendations

1. Introduction

Guideline

This section specifies the organisational framework within which the process assessment has been performed.

2. Assessment Context

The objective of this section is to provide all information needed to understand in which context the process assessment was performed. Context information is mandatory for a SPICE conformant assessment. Context information allow readers to evaluate when, by whom, with regards to which objectives the results of the process assessment where derived. Context information includes:

- The assessment purpose
- Date when the assessment was performed
- The assessment scope, including target organisational units, processes to assess and reference documentation used to support the evaluation
- Assessment constraints
- Assessment responsibilities

Guideline

Complete each bullet with the relevant information. Most of this information can be derived directly from the assessment plan.

3. Assessment Results

Guideline

Assessment results include profiles of the assessed processes (both capability level profiles and process attribute profiles for SPICE conformant assessments) and detailed findings for each process based on interview results and analysis of relevant documentation.

4. Strengths and Weaknesses

Guideline

This section identifies strengths and weaknesses in the assessed organisational units. Strengths and weaknesses emerge from analysis of the current capability of assessed processes against the identified organisation objectives.

5. Improvement Recommendation

Guideline

Processes recommended for improvement are identified. For each process suggestions should be provided on actions that should be performed to improve the process. Improvement actions may include establishment of missing practices, adoption of methods and tools, preparation of working instructions, training.

Suggestions on improvement priorities should be provided as well. Priorities will depend on several factors, including evaluation of most urgent needs, feasibility of improvement actions, time and cost constraints.

A table can be used to show the level of priority based on evaluation of each process capability against process impact on the organisation's objectives. An example of such a table is as follows:

GOALS and PROCESSES			CAPABILITY		PRIORITY
Goal	Ranking	Related process	Organization	Project	
<ISO 9126 extended Characteristic/ sub-characteristic>	<Ranking value>	<Process n>	<Assessed capability>	<Assessed capability>	<Priority>
<ISO 9126 extended Characteristic/ sub-characteristic>	<Ranking value>	<Process m>	<Assessed capability>	<Assessed capability>	<Priority>
	<Ranking value>	<Process x>	<Assessed capability>	<Assessed capability>	<Priority>
<ISO 9126 extended Characteristic/ sub-characteristic>	<Ranking value>	<Process y>	<Assessed capability>	<Assessed capability>	
<ISO 9126 extended Characteristic/ sub-characteristic>	<Ranking value>	<Process z>	<Assessed capability>	<Assessed capability>	<Priority>



APPENDIX 2.4

THE PROFES TEMPLATE FOR:

PROCESS ASSESSMENT EFFORT COLLECTION

Guideline

Collecting effort information about the actual improvement cycle allows to perform better estimates in the future and is necessary to perform an evaluation of the improvement costs against improvement benefits. The following template is designed to support effort data collection for a BOOTSTRAP process assessment. The template include two main parts:

The first page is designed to provide information on how each assessment day was used, including information of assessment activities performed, their duration, participating roles and number of people involved.

The second page is designed to provide detailed effort information for each assessment activity and for each role.

A list of roles and activities is provided.

Assessment: Actual Daily Schedule

Date of assessment:

Assessment at site:

Name of data provider:

Date of data providing:

Please indicate the actual course of activities during the assesement day by marking the time span of each working session. Then supply the names of activities as well as names and numbers of the participating roles.

For information on terminology please refer to the appendix of this collection of data collection sheets.

Time	Session	Activities	Roles	Number of People
08.00				
09.00				
10.00				
11.00				
12.00				
13.00				
14.00				
15.00				
16.00				
17.00				
18.00				
19.00				
20.00				
21.00				
22.00				

List of Activities and Roles

Activities

- Preparation (all activities from "pre-assessment briefing" to "execution scheduling")
code: Prep
- Opening briefing
code: O_Brief
- Global site assessment (SPU)
code: Asmt_S
- Project assessment (project)
code: Asmt_P
- Evaluation
code: Eval
- Project Assessment review
code: Rev
- On-site final meeting
code: Fin_Meet
- Prepare Assessment report
code: Rep_Prep
- Review Assessment report
code: Rep_Rev
- Other
code: O - please indicate which other activity

Roles

- Lead Assessor (L)
- Assessor (A)
- Manager (M)
- Software Engineer (E)
- Facilitator (F)
- Other (O) - if: indicate which other role

Note: purely observing assessors are not reported



APPENDIX 2.5

THE PROFES TEMPLATE FOR: PPD MODEL

Guideline

The following questionnaire can be used to elicit PPDs.

The input to the questionnaire is defined as follows:

Technology	Describe the practice, method or tool applied
Process	Identifies the process to which the technology contributes
Product/Product Type	Identifies the product or product type to which the PPD applies
Quality	Specify the quality characteristic that is affected by the technology. The quality can be described using ISO 9126 extended characteristics and sub-characteristics
CF	Context Factor. Describe one specific context factor like size and duration of the project, development language used, experience of engineers etc.

Information provider	Name of the engineer that provides the PPD
Affiliation	Organisational unit of the information provider
Interviewer	Name of the expert that has interviewed the information provider
Date	Date of interview or generation of the PPD
PPD Identification	PPD identifier in the PPD repository

Guideline

Use this template during PPD interviews or for capturing PPD-related information in assessment interviews or GQM interviews. Write the technology, process, and product quality in the header of the form. For making notes about context factors (*CF*), use the ten boxes and lines in the middle of the form. Don't forget to make notes about the information provider and the date at which the information was acquired. Use the box at the bottom of the form for these notes.

For filling in information about a context factor, write the name of the context factor in one of the boxes, indicate the possible values of the context factor on the associated line, and mark those values that are required for the effective application of the technology (i.e., *application domain*). For numeric value ranges (e.g., years of experience or project team size in terms of number of persons), write the minimum and maximum values at the ends of the line and mark the application domain as a range on the line. For ordinal and nominal values (e.g., low/average/high or the names of programming languages), write down each value on the line. Mark the application domain by underlining or by drawing circles around the relevant values.

After the interview, transfer the information into the PPD repository. For possible later needs for clarification, keep the interview notes and indicate on the filled-in form the identifiers of the derived PPDs in the repository.

Technology	
Process	
Product/Product Type	
Product Quality	

Technology Application Context	
---------------------------------------	--

Context Factors	Context Characterisation
-----------------	--------------------------

CF.1		
CF.2		
CF.3		
CF.4		
CF.5		
CF.6		
CF.7		
CF.8		
CF.9		
CF.10		

Information Provider	
Affiliation	
Interviewer	
Date	
PPD Identification	



APPENDIX 2.6

THE PROFES TEMPLATE FOR: PROCESS IMPROVEMENT PLAN

Table of Contents

1. Introduction
 - 1.1 Purpose
 - 1.2 Scope
 - 1.3 Document Description
 - 1.4 Document Update Process
 - 1.5 References
 - 1.6 Terms and Definitions
2. SPU Improvement Plan
 - 2.1 Product Improvement Goals
 - 2.1.1 Final Product Improvement Goals
 - 2.1.2 Software Improvement Goals
 - 2.2 Process Improvement Goals

- 2.3 Detailed Description of Process Changes and Improvements
- 2.4 Needs
- 2.5 Organisation to Support Improvement
- 2.6 Improvement Validation
- 2.7 Improvement Phases and Activities
- 3. Project <project name> Improvement Plan
 - 3.1 Product Improvement Goals
 - 3.1.1 Product Improvement Goals
 - 3.1.2 Software Improvement Goals
 - 3.2 Process Improvement Goals
 - 3.3 Detailed Description of Process Changes and Improvements
 - 3.4 Needs
 - 3.5 Organisation to Support Improvement
 - 3.6 Improvement Validation
 - 3.7 Improvement Phases and Activities
- 4. Budget
- 5. Schedule

1.1 Introduction

1.1 Purpose

The purpose of this Process Improvement Plan (PIP) is to describe the improvement activities to be carried out at <SPU name> and <project name> for the period <period definition>.

This PIP is based on findings from <reference any of the following: process assessment report, product assessment report, report on measurements collected in a defined period>.

1.2 Scope

Guideline

This PIP can be applied to the SPU and/or one or more projects. Provide an unambiguous reference to both the SPU and/or the project(s) to which this PIP is applicable. This PIP focuses on achievement of specified product quality goals. Provide an unambiguous reference to the products to which this PIP is applicable. At SPU level quality goals and quality improvement goals can be referred to a single product or to a product family. At project level quality goals and quality improvement goals can be referred to a single product or to a product subsystem/component. If improvement is planned both at SPU and project level, specify how they relate to each other.

1.3 Document Description

This PIP is structured into the following main parts:

Guideline

A description of main chapters and annexes should be included, chapter per chapter, annex per annex.

1.4 Document Update Process

This PIP includes the initial schedule for the suggested improvement actions. The document will be progressively completed and updated following the progressive detailed specification and organisation of the planned improvement actions.

Guideline

It is important to:

- establish Progress Status Report schedule
- define how to manage schedule updating for planned activities
- define how to manage the versions of this document

1.5 References

Guideline

Include here reference to all the documents related to this PIP.

- [1] < Title of the applicable process assessment report >
<document code>
- [2] < Title of the applicable product assessment report >
<document code>
- [3] <Title of the applicable GQM plan>
<document code>
- [4] <Title of the applicable GQM measurement plan>
<document code>
- [5] < Title of the applicable measurement report >
<document code>
- [6] < Title of the applicable descriptive process model>
<document code>
- [7] < Title of the applicable prescriptive process model>
<document code>
- [8] < Title of the applicable project plan>
<document code>
- [9] ISO/IEC 15504: Information Technology - Software Process Assessment

1.6 Terms and definitions

PSR Progress Status Report

PIP Process Improvement Plan

GQM	Goal/Question/Metric
PPD	Product-Process Dependency.
DPM	Descriptive Process Model. This is the process model that describes how a process is actually carried out in an organisation
PDM	Prescriptive Process Model. This is a process model and/or procedure that provides directives on how to carry out a process
QIP	Quality Improvement Paradigm
SPU	Software Producing Unit

2. SPU IMPROVEMENT PLAN

2.1 Product Improvement Goals

2.1.1 Final Product Improvement Goals

Guideline

Include here a description of the quality goals and quality improvement goals of the final product/product family as agreed with the final customer/user or declared as product standard performances in marketing documentation. Reference the applicable GQM plan and list the applicable questions. Reference the applicable measurement plan and provide a list of applicable metrics and related target quantitative value.

2.1.2 Software Improvement Goals

Guideline

Include here a description of the software quality goals and quality improvement goals as deployed from the overall product quality goals and improvement goals. Reference the applicable GQM plan and list the applicable questions. Reference the applicable GQM measurement plan and provide a list of applicable metrics and related target quantitative value.

2.2 Process Improvement Goals

Guideline

Include here a description of the process goals and process improvement goals that affect achievement of the product and/or software goals. Reference the applicable assessment report; list the suggested process improvements and target capability to be achieved for each process as recommended in the process assessment report. Reference the applicable PPDs (either hypothetical or already validated). Reference the applicable GQM plan and list the applicable questions. Reference the applicable GQM measurement plan and provide a list of applicable metrics and related target quantitative value.

2.3 Detailed Description of Process Changes and Improvements

Guideline

This section provides an operational description of planned process changes and improvements. For each process the following should be specified:

- which changes/improvements are planned. Changes may affect either organisational directives on how to perform a process (prescriptive process model) or the way how a process is carried out in daily routine (descriptive process model) or both. Changes and improvements may include some of the following: implementation of missing practices (base practices) and/or improvement of the way how a process is managed (for instance how a process is planned and controlled - management practices); adoption of methods and tools to support specific processes and/or practices ; change in roles and responsibilities. Mention here any applicable reusable experience, if available.
- which activities/actions are planned to implement the identified changes/improvements. It should be defined whether any preliminary analysis/piloting is needed. Supporting actions as briefings, training etc. should be also specified.
- which methods and/or tools are going to be evaluated or adopted (if this is already known)
- who will be responsible for each activity/action

2.4 Needs

Guideline

The purpose of this section is to identify any need to be satisfied in order to complete the defined improvement actions. Examples of needs are as follows: technical information to be achieved, management decisions, availability of resources, training etc.

2.5 Organisation to Support Improvement

Guideline

The purpose of this section is to define roles, responsibilities and the organisation for improvement. Particularly the following roles should be established:

- the responsibility to co-ordinate and manage planned improvements
- the staff to implement the improvement actions and to package experience
- management representative
- the role of consultants (if any)

2.6 Improvement Validation

Guideline

The purpose of this section is to define how the selected product and process improvements will be validated at SPU level. Recommended methods to validate improvement are:

- repetition of a process assessment, possibly just on the processes affected by the implemented improvement actions
- collection and analysis of measurement data as specified in the applicable GQM plan and measurement plan. Product measurement should be collected.

2.7 Improvement Phases and Activities

Guideline

The objective of this section is to describe which activities will be performed to carry out the selected improvements. It is suggested to use the PROFES phases and steps as a guideline. This PIP is expected to be developed during the PROFES phase "Set Goals": nevertheless it is recognised that in real cases activities may proceed in parallel or anyway may be organised according to specific needs so that sequence of activities may be slightly different. Therefore all PROFES phases may need to be addressed in this plan. The PROFES improvement methodology can be used as a guideline or a checklist of suggested activities to be performed. Complete the following text as needed:

Phase 1. Characterize

Describe which activities will be performed to fulfill the following phase steps:

1. Verify commitment
2. Identify product quality needs
3. Determine current product quality
4. Determine current process capability

Provide a schedule of such activities.

Phase 2: Set Goals

The results of the following PROFES steps will be documented in this plan:

6. Set Product improvement goals
7. Determine necessary process changes

Phase 3: Plan

The following PROFES steps will be planned and specified in this plan:

8. Describe process changes
9. Set metrics for the processes and product

Detailed process changes will be documented in a (prescriptive) process model. Metrics for the product and processes will be documented in the GQM and measurement plan.

Phase 4: Implement and monitor improvement

Specify which activities will be performed to carry out the planned improvement actions (Step 10. Implement and monitor improvements). Specify additionally how measurement data will be collected and feedback sessions performed and reference the applicable GQM plan and measurement plan if already available. Specify whether process or product re-assessment is planned.

Phase 5: Analyse

Specify activities to analyse improvement results (Step 11. Evaluate results).

Phase 6: Package

Specify activities to update the experience base (Step 12. Update experience base).

3. PROJECT<project name> IMPROVEMENT PLAN

3.1 PRODUCT Improvement Goals

3.1.1 Product Improvement Goals

Guideline:

Include here a description of the quality goals and quality improvement goals of the product or product component as agreed with the final customer/user/project orderer or declared as product standard performances in marketing documentation. Reference the applicable GQM plan and list the applicable questions. Reference the applicable GQM measurement plan and provide a list of applicable metrics and related target quantitative value.

3.1.2 Software Improvement Goals

Guideline:

Include here a description of the quality goals and quality improvement goals of the software that will be developed in the project. Software quality goals and improvement goals should be deployed from the quality goals and improvement goals of the overall product or product component. Reference the applicable GQM plan and list the applicable questions. Reference the applicable GQM measurement plan and provide a list of applicable metrics and related target quantitative value.

3.2 Process Improvement Goals

Guideline:

Include here a description of the process goals and process improvement goals to be achieved in the project. Reference the applicable assessment report; list the suggested process improvements and target capability to be achieved for each process as recommended in the process assessment report. Reference the applicable PPDs (either hypothetical or already validated). Reference the applicable GQM plan and list the applicable questions. Reference the applicable GQM measurement plan and provide a list of applicable metrics and related target quantitative value.

3.3 Detailed Description of Process Changes and Improvements

Guideline:

This paragraph provides an operational description of planned process changes and improvements. For each process the following should be specified:

- which changes/improvement are planned. Changes and improvements may include some of the following: tailoring of the applicable prescriptive process model or definition of a specific process model for the project; implementation of missing practices (base practices) and/or improvement of the way how a process is managed (for instance how a process is planned and controlled - management practices); adoption of methods and tools to support specific processes and/or practices ; change in roles and responsibilities. Mention here any applicable reusable experience, if available.
- which activities/actions are planned to implement the identified changes/improvements. It should be defined whether any preliminary

analysis/piloting is needed. Supporting actions as briefings, training etc. should be also specified.

- which methods and/or tools are going to be evaluated and/or adopted (if this is known already)
- who will be responsible for each activity/action

3.4 Needs

Guideline:

The purpose of this section is to identify any need to be satisfied in order to complete the defined improvement actions. Examples of needs are as follows: technical information to be achieved, management decisions, availability of resources, training etc.

3.5 Organisation to Support Improvement

Guideline:

The purpose of this section is to define roles, responsibilities and the organisation for improvement. Particularly the following roles should be established:

- the responsibility to co-ordinate and manage planned improvements, including definition of reporting towards SPU improvement responsibilities (if any) and relationships with project management
- the staff to implement the improvement actions and to package experience.
- management representative
- the role of consultants (if any)

3.6 Improvement Validation

Guideline:

The purpose of this section is to define how the selected product and process improvements will be validated respectively at project level and SPU level (if applicable). Recommended methods to validate improvements are:

- repetition of a process assessment, possibly just on the processes affected by the implemented improvement actions

- collection and analysis of measurements as specified in the applicable GQM plan and measurement plan. Data on product improvements should be collected.

3.7 Improvement Phases and Activities

Guideline:

The objective of this section is to describe which activities will be performed at project level to carry out the selected improvements. It is suggested to use the PROFES phases and steps as a guideline. This PIP is expected to be developed during the PROFES phase "Set Goals": nevertheless it is recognised that in real cases activities may proceed in parallel or anyway may be organised according to specific needs so that sequence of activities may be slightly different. Therefore all PROFES phases may need to be addressed in this plan. The PROFES improvement methodology can be used as a guideline or a checklist of suggested activities to be performed. Complete the following text as needed:

Phase 1. Characterize

Describe which activities will be performed to fulfill the following phase steps:

1. Verify commitment
1. Identify product quality needs
2. Determine current product quality
3. Determine current process capability

Provide a schedule of such activities.

Phase 2: Set Goals

The results of the following PROFES steps will be documented in this plan:

5. Set Product improvement goals
6. Determine necessary process changes

Phase 3: Plan

The following PROFES steps will be planned and specified in this plan:

7. Describe process changes

8. Set metrics for the processes and product

Detailed process changes will be documented in a (prescriptive) process model. Metrics for the product and processes will be documented in the GQM and measurement plan.

Phase 4: Implement and monitor improvement

Specify which activities will be performed to carry out the planned improvement actions (Step 10. Implement and monitor improvements). Specify additionally how measurement data will be collected and feedback sessions performed and reference the applicable GQM plan and measurement plan if already available. Specify whether process or product re-assessment is planned.

Phase 5: Analyse

Specify activities to analyse improvement results (Step 11. Evaluate results).

Phase 6: Package

Specify activities to update the experience base (Step 12. Update experience base).

4. Budget

Guideline:

Specify budgeted costs for the planned improvements, both at organisational and project level, including:

- labour
- licenses to be purchased (if any)
- additional equipment (if any)
- training costs
- costs of consulting services (if any)

5. Schedule

Guideline:

This section should include the schedule and the effort needed to perform the above mentioned improvement actions at SPU and project level as applicable. A Gantt chart should be included. This schedule will be the initial reference for improvement progress monitoring and will be updated as necessary.



THE PROFES TEMPLATE FOR:

GQM Plan

Table of Contents

1. Introduction
2. Product and Project Description
 - 2.1 Product Related Information
 - 2.1.1 Product
 - 2.1.2 Product Family
 - 2.1.3 Role of Software in the Product
 - 2.1.4 Product within the Measurement Programme
 - 2.2 Project Related Information
 - 2.2.1 Project Organisation
 - 2.2.2 Project Duration
 - 2.2.3 Project Development Process

2.2.4 Project Status

3. Measurement Goals
4. Abstraction Sheets
5. GQM Plan for Each Measurement Goal

1. Introduction

Guideline

This section specifies the organisational framework and context within which this GQM plan has been prepared.

2. Product and Project Description

Guideline

The objective of this section is to provide a synthetic description of the target product and project for the GQM plan. The following is a suggested structure for this section.

2.1 Product Related Information

2.1.1 Product

2.1.1.1 Product Family

2.1.1.2 Role of Software in the Product

2.1.1.3 Product within the Measurement Programme

2.1.2 Project Related Information

2.1.2.1 Project Organisation

2.1.2.2 Project Duration

2.1.2.3 Project Development Process

2.1.2.4 Project Status

3. Measurement Goals

Practice has shown the importance of specifying a measurement goal precisely as the selection and definition of suitable and useful measures and models depends strongly on the clarity of these early decisions. GQM provides a template for defining measurement goals in a precise way.

GQM Goal template structures a measurement goal according to five aspects:

- The *object of study* defines the primary target of the study, i.e., the process or product that will be analysed. Examples of objects are the entire development process, individual process phases (e.g., system test), documents (e.g., design documents), or the final product.
- The *purpose* of the study expresses why the object shall be analysed. Common purposes are the following:
 - *Characterisation* aims at providing a snapshot of the current state/performance of the development processes and products.
 - *Monitoring* aims at following the trend/evolution of the current performance/state of the development processes and products.
 - *Evaluation* aims at comparing and assessing the quality of products and the efficiency/effectiveness of processes.
 - *Prediction* aims at identifying relationships between various process and product attributes (influencing factors), and using these relationships to predict external attributes of products and processes.
 - *Control and change* aim at identifying causal relationships that influence the state/performance of processes and products. Control consists in influencing the course of a project in order to alleviate risks. Change consists in modifying the development process from project to project in order to improve quality or productivity. Change requires a finer grained understanding of the phenomena under study than control.
- The *quality focus* states the particular attribute of the object of study that shall be characterised, evaluated, predicted, monitored, controlled, or changed. Examples for quality focuses are cost, reliability, correctness, defect removal, modification, user friendliness, maintainability, etc.
- The *viewpoint* identifies the roles or positions of the individuals who are going to use the output of the measurement programme, for example, those who interpret the data collected and use the models derived from

the data. Examples for viewpoints are project leader, developer, system tester, quality assurance manager, user, etc.

- The *context* of the study specifies the environment in which the study will be performed, and correspondingly determines to what extent the results can be generalised. The information contained in the context is used to make environment-specific influencing factors explicit, for example, experience of the developers, or application domain of the product.

Guideline

A GQM measurement goal is described using a table format as follows. Complete the following table format for each GQM goal you have identified.

Object	
Purpose	
Quality Focus	
Viewpoint	
Context	

Guideline

The meaning of each entry in the table is described as follows:

Object (of Study)	What will be analysed? Examples: development process, system test, design document, end product, etc.
Purpose	Why will the object be analysed? Examples: characterisation, monitoring, evaluation, prediction, control, change
Quality Focus	What property/attribute of the object will be analysed? Examples: reliability, maintainability, cost, correctness, defect removal, modification, user friendliness, etc.

Viewpoint	Who will use the collected data? Examples: project leader, developer, system tester, quality assurance manager, user, senior management, etc.
Context	In which environment will the measurement programme be performed? Examples: project X, company Y, department Z, etc.

Guideline

- Any measurement goal should be expressed with the help of the GQM Goal Template.
- A measurement goal should not cluster more than one purpose, quality focus, or viewpoint. Even though similar data might be collected, clustering is likely to create confusion.

Example 1:

Analyse the end product
for the purpose of characterisation
with respect to reliability
from the viewpoint of the system tester
in the context of project X

Object	<i>End Product A</i>
Purpose	<i>Characterise</i>
Quality Focus	<i>Reliability</i>
Viewpoint	<i>System Tester</i>
Context	<i>Project X</i>

Example 2:

**Analyse the system test
for the purpose of prediction
with respect to effectiveness
from the viewpoint of the system tester
in the context of organisation A**

Object	<i>System Test</i>
Purpose	<i>Prediction</i>
Quality Focus	<i>Effectiveness</i>
Viewpoint	<i>System Tester</i>
Context	<i>Organisation A</i>

4. Abstraction Sheets

An Abstraction Sheet helps to elicit and structure the current knowledge of individuals with respect to a precisely defined measurement goal. Abstraction Sheets can be used for two purposes:

- **Development of GQM Plans:** Individuals that have been identified by the viewpoint in the measurement goal definition are asked to share their knowledge and expectations with the measurement analyst. For that purpose GQM interviews are conducted during which Abstraction Sheets are filled-in. For the development of a GQM Plan, the information contained in Abstraction Sheets from different interview sessions is merged, consolidated and formalised.
- **Interpretation and analysis of measurement data:** during the execution of a measurement programme the measurement analyst conducts so-called feedback sessions on a regular base to inform about and interpret the results of the data analyses performed. Using the aggregated knowledge representation structures provided by the Abstraction Sheets can better focus the interpretation of analysis results.

GQM Plans are constructed by defining and combining questions, models, and measures on the *viewpoints'* experience. For each measurement goal one GQM Plan will be developed. The *viewpoint* - as defined in a measurement goal - does not need to see all the details of a GQM plan. A GQM Plan is constructed by the measurement analyst based on a viewpoint's experience. To support the structured interaction of the measurement analyst with the viewpoint, the Abstraction Sheet template was developed. An Abstraction Sheet is a template that supports knowledge acquisition when developing a GQM Plan

In order to capture the experience of a viewpoint, the GQM Abstraction Sheets are used as a knowledge acquisition instrument during interviews. An Abstraction Sheet contains information about the entities of the measurement object with its associated attributes representing the quality focus (as specified by the measurement goal), and information about factors that have an impact on the quality focus (so-called variation factors). In addition, hypotheses about the performance of the quality focus attributes and the way in which the variation factors influence the performance of the quality focus attributes are documented. Based on this information, for each measurement goal, a set of questions, measures, and models can easily be defined. The components of Abstraction Sheets, referred to as quadrants, cover the essential topics that the viewpoint needs to address during interviews. In the following, the content of each quadrant is described:

- **Quality Focus:** This quadrant captures information that defines the quality focus on the object as specified in the measurement goal. The information is intended to capture the viewpoint's intuition about the quality focus and transforms it into an operational definition (through measures).
- **Baseline Hypotheses:** This quadrant captures the expectations a viewpoint has with respect to the measures defining the quality focus.
- **Variation Factors:** This quadrant captures the factors that, according to the viewpoint, have an impact on the quality factor in a particular context. These factors will trigger the formulation of questions and definition of related measures
- **Impact on Baseline Hypotheses:** In this quadrant the expected impact of the variation factors on the quality focus are captured. For each variation factor an explicit relationship to the quality focus must be established. To avoid that GQM Plans become too complex, in this quadrant only those hypotheses should be noted that can be supported by the viewpoint's experience to some reasonable degree.

Guideline

Complete the following abstraction sheet:

Measurement Goal				
Object	Purpose	Quality focus	Viewpoint	Environment
<...>	<...>	<...>	<...>	<...>

Quality focus	Variation factors
<p><i>Which factors define the quality focus?</i></p> <p><u>Process (or Product) Definition:</u></p> <p>Question Q-QF1 <...> Measure M-QF1.1 <...> Measure M-QF1.2 <...> ... Measure M-QF1.n₁ <...></p> <p>Question Q-QF2 <...> Measure ...</p> <p>...</p> <p>Question Q-QFm <...> Measure ...</p>	<p><i>Which factors have an impact on the quality focus?</i></p> <p><u>Process Conformance:</u></p> <p>Question Q-VF1 <...> Measure M-VF1.1 <...> Measure M-VF1.2 <...> ... Measure M-VF1.p₁ <...></p> <p>Question Q-VF2 <...> Measure ...</p> <p>...</p> <p>Question Q-VF_{k_{conf}} <...> Measure ...</p> <p><u>Process Domain Understanding:</u></p> <p>Question Q-QF_{k_{conf}+1} <...> Measure ...</p> <p>...</p> <p>Question Q-QF_{k_{under}} <...> Measure ...</p> <p><u>Other:</u></p> <p>Question Q-QF_{k_{under}+1} <...> Measure ...</p> <p>...</p> <p>Question Q-QF_{k_i} <...> Measure ...</p>
Baseline hypotheses	Impact on baseline hypotheses
<p><i>What is the current expectation wrt. the quality focus ?</i></p> <p>Hypothesis H-QF1.1 <...> Hypothesis H-QF1.2 <...> ... Hypothesis H-QF1.i₁ <...> Hypothesis H-QF2.1 <...> ... Hypothesis H-QFm.im <...></p>	<p><i>How do the variation factors influence the quality focus ?</i></p> <p>Hypothesis H-QF1.1 <...> Hypothesis H-QF1.2 <...> ... Hypothesis H-QF1.j₁ <...> Hypothesis H-QF2.1 <...> ... Hypothesis H-QF_{k_i}.j_{k_i} <...></p>

Example Abstraction Sheet:

Measurement Goal				
Object	Purpose	Quality focus	Viewpoint	Environment
System Test	Predict	Effectiveness	System Tester	Organisation A

Quality focus	Variation factors
<i>Which factors define the quality focus?</i>	<i>Which factors have an impact on the quality focus?</i>
<p><u>Process (or Product) Definition:</u> Q-QF1 How many failures (classified) will occur during field test? M-QF1.1 Number of failures during system test M-QF1.2 Failure class (criticality: critical, uncritical) Q-QF2 How many faults (classified) will be detected in the software during system test? M-QF2.1 Number of faults detected during system test M-QF2.2 Fault class (severity: fatal, major, minor)</p>	<p><u>Process Conformance:</u> Q-VF1 How good is the average quality of test cases? M-VF1.1 Number of test cases M-VF1.2 Quality of test case: low, medium, high M-VF1.3 Degree of code coverage with test cases: covering of paths, statements, nodes, etc. in percent Q-VF2 What test methods are used? M-VF2.1 Test method: a, b, c, ... Q-VF3 How well is a test method performed? M-VF3.1 Test method conformance: low, medium, high</p> <p><u>Process Domain Understanding:</u> Q-VF4 How experienced are the testers wrt. the testing tools? M-VF4.1 Experience of testers: insufficient, sufficient, excellent Q-VF5 How well do the testers understand the requirements? M-VF5.1 Understandability of requirements documents: good, bad. M-VF5.2 Size of requirements documents (pages) Q-VF6 How well do the testers understand the source code? M-VF6.1 Understandability of source code: good, bad M-VF6.2 Size of source code modules (lines of code) M-VF6.3 Complexity of source code modules (McCabe complexity) M-VF6.4 Mastery of used programming languages: insufficient, sufficient, excellent</p>
Baseline hypothesis	Impact on baseline hypothesis
<i>What is the current expectation wrt. the quality focus?</i>	<i>How do the variation factors influence the quality focus?</i>
<p>H-QF1.1: 10 failures of class "critical" will occur (33%). H-QF1.2: 20 failures of class "uncritical" will occur (67%).</p>	<p>H-VF1.1: The higher the quality of the test cases is, the more failures will be detected. H-VF2.1: Different testing methods detect different types and numbers of failures.</p>

<p>H-QF2.1: 8 defects of class "fatal" will be detected (20%). H-QF2.2: 24 faults of class "major" will be detected (60%). H-QF2.3: 8 faults of class "minor" will be detected (20%).</p>	<p>H-VF3.1: The better a test method conformance is, the more failures will be detected. H-VF4.1: The higher the experience with the testing tools, the more failures will be detected. H-VF5.1: The better the understandability of the requirements, the more failures will be detected. H-VF6.1: The better the understandability of the source code, the more faults will be located.</p>
---	--

5. GQM Plans for Each Measurement Goal

The GQM Plan is the result of merging the available Abstraction Sheets that were filled-in during GQM interview sessions. For each measurement goal, a separate GQM Plan should be set-up.

Guideline

The template for the GQM plan is as follows. Complete the following template for each measurement goal.

Document-ID <GQM-Plan Nr. X>	
Measurement Goal Specification <using the GQM Goal Template>	
<p><i>Analyse the <object> for the purpose of <characterisation, monitoring, evaluation, prediction, control/change> with respect to <quality focus> from the viewpoint of <role, organisation> in the context of <environment></i></p>	
Quality Focus <using GQM Abstraction Sheets>	
Process (or Product) Definition	
Question Q-QF1 <...>	
	Measure M-QF1.1 <...> Measure M-QF1.2 <...> ... Measure M-QF1.n ₁ <...>
Question Q-QF2 <...>	
	Measure M-QF2.1 <...> Measure M-QF2.2 <...> ... Measure M-QF2.n ₂ <...>
...	
Question Q-QFm <...>	
	Measure M-QFm.1 <...> Measure M-QFm.2 <...> ... Measure M-QFm.n _m <...>
Variation Factors <using GQM Abstraction Sheets>	

Process Conformance	
Question Q-VF1 <...>	
	Measure M-VF1.1 <...> Measure M-VF1.2 <...> ... Measure M-VF1.p ₁ <...>
Question Q-VF2 <...>	
	Measures ...
...	
Question Q-VF _{conf} <...>	
	Measures ...
Process Domain Understanding	
Question Q-QF _{conf+1} <...>	
	Measures ...
...	
Question Q-QF _{under} <...>	
	Measures ...
Other	
Question Q-QF _{under+1} <...>	
	Measures ...
...	
Question Q-QF _t <...>	
	Measures ...
Baseline Hypotheses <using GQM Abstraction Sheets>	
	Hypothesis H-QF1.1 <...> Hypothesis H-QF1.2 <...> ... Hypothesis H-QF1.i1 <...>
	Hypothesis H-QF2.1 <...> ...
...	
	Hypothesis H-QFm.im <...> ...
Impact on Baseline Hypotheses <using GQM Abstraction Sheets>	

Hypothesis H-QF1.1 <...> Hypothesis H-QF1.2 <...> ... Hypothesis H-QF1.j1 <...>
Hypothesis H-QF2.1 <...> ...
...
Hypothesis H-QFk _i .jk _i <...> ...



APPENDIX 2.8

THE PROFES TEMPLATE FOR: MEASUREMENT PLAN

Table of Contents

1. Introduction
 - 1.1 Audience
 - 1.2 Measuring Period
 - 1.3 Measurement Scope
 - 1.4 Terms and Abbreviations
 - 1.5 Process Model
2. Measurement Goals
3. Measured Metrics
4. Measurement Procedures
5. Data Collection Forms
6. Measurement Database

1. Introduction

The purpose of the measurement plan is to:

- specify in brief *when* and *how* the data are collected
- define *who* is the provider of data
- describe how data are collected
- describe *where* the data is fetched (database/tool)

1.1 Audience

This document is to be used by <provide a list of intended readers and users of this document>.

1.2 Measuring period

Guideline

This section will specify when the measurement data will be collected.

1.3 Measurement Scope

The initial measurement will be restricted to <specify the organisational boundaries within which measurement data will be collected>.

1.4 Terms and Abbreviations

GQM

Goal/ Question/ Metric

1.5 Process model

Guideline

Reference the applicable process model that specifies the phases and activities during which data will be collected.

2. Measurement goals

The measurement goals are derived from the following product and process improvement goals:

Product improvement goal	Process improvement goal

3. Measured metrics

Guideline

For each metric the following should be defined:

- a formal definition of the metric
- a textual explanation of the metric
- the range (possible values) of the metric
- the role (e.g. engineer, manager) that collects the data
- when data are to be collected
- the medium by which data are collected (e.g. tools, data collection forms)

The following table can be used:

<i>Column Heading</i>	<i>Description</i>
Id	Unique identifier of the metric.
Measure Name	Brief textual description used to verbally identify the measure.
Definition	Short but exact measurement definition. Description of what is measured by the measure.
Range of Values	Possible range of values
Scope	Organisational unit where data are collected
Trigger	Event(s) showing when the measure is available and can be collected.
Provider/Source	Person providing raw measurement data or tool/database that is to be used if data already exists.
Form/Tool	Method of data collection. If data are obtained using specific forms or special questionnaires, these are stated here.
Responsible	Person responsible for collection and aggregation of raw data.

Such a table should be created for each metric defined in the GQM plan.

4. Measurement procedures

Guideline

This section specifies the data collection procedures.

5. Data Collection Forms

Guideline

This section provides all data collection forms that will be used to collect measurement data.

6. Measurement Database

Guideline

The database is used to store the data collected from multiple sources. It should provide the following functionality:

- Display of all raw data.
- Advanced search functions.
- Statistical analysis of all data.
- Preparing presentation of results.

- Easy data import and export functions to allow porting data to and from other tooling.

The following page shows an example of a database schema (with references to questions and metrics definitions provided in the related GQM plan).



APPENDIX 2.9

THE PROFES TEMPLATE FOR: GQM EFFORT COLLECTION

Guideline

Collecting effort information about the actual improvement cycle allows to perform better estimates in the future and is necessary to perform an evaluation of the improvement costs against improvement benefits. The following template is designed to support effort data collection for GQM measurement activities. The template include two main parts:

The first page is designed to provide information on how each day was used, including information of GQM activities performed, their duration, participating roles and number of people involved.

The second page is designed to provide detailed effort information for each GQM activity and for each role.

A list of roles and activities is provided.

GQM-Based Measurement: Actual Daily Schedule

Date of reported day:

Application site:

Name of data provider:

Date of data providing:

Please indicate the actual course of activities during the GQM-related activities of the reported day by marking the time span of each working session. Then supply the names of activities as well as names and numbers of the participating roles.

For information on terminology please refer to the appendix of this collection of data collection sheets.

Time	Session	Activities	Roles	Number of People
08.00				
09.00				
10.00				
11.00				
12.00				
13.00				
14.00				
15.00				
16.00				
17.00				
18.00				
19.00				
20.00				
21.00				
22.00				

List of Activities and Roles

Overall and detailed activities

- Prepare Measurement Programme
 - Activities are:
 - Identify available input, preconditions and constraints (PMP_I)
 - Set up infrastructure and select project (PMP_S)
 - Plan measurement programme (PMP_P)
 - Prepare and conduct training of GQM facilitators (PMP_T)
 - Other (PMP_O)
- Identify and Define GQM Goals
 - Activities are:
 - Characterise project and organisation (IDG_C)
 - Identify and select improvement goals (IDG_I)
 - Define measurement and GQM goals (IDG_D)
 - Identify relevant artifacts that can possibly be reused (IDG_R)
 - Other (IDG_O)
- Prepare and Conduct GQM Interviews
 - Activities are:
 - Study material and documentation (PCI_S)
 - Define schedule and invite interviewees (PCI_D)
 - Conduct briefing of project team (PCI_C)
 - Hold GQM interviews and create abstraction sheets (PCI_I)
 - Other (PCI_O)
- Develop GQM Plan
 - Activities are:
 - Define GQM plan (DGP_D)
 - Review GQM plan (DGP_R)
 - Refine GQM plan (DGP_G)
 - Other (DGP_O)

- Develop Measurement Plan
 - Activities are:
 - Define measurement plan (DMP_D)
 - Identify and define data collection procedures (DMP_C)
 - Review measurement plan and data collection procedures (DMP_R)
 - Refine measurement plan (DMP_M)
 - Other (DMP_O)
- Perform Data Collection
 - Activities are:
 - Conduct briefing of participants and kick-off of measurement programme (PDC_B)
 - Collect data (PDC_C)
 - Validate data (PDC_V)
 - Code and store data (PDC_S)
 - Other (PDC_O)
- Perform Data Analysis and Interpretation
 - Activities are:
 - Analyze data (PAI_A)
 - Prepare presentation material (PAI_P)
 - Plan feedback sessions and invite participants (PAI_F)
 - Conduct feedback sessions (PAI_C)
 - Other (PAI_O)
- Package Experience
 - Activities are:
 - Package measurement data (PE_M)
 - Package results of analysis and interpretation (PE_R)
 - Store experience packages (PE_S)
 - Other (PE_O)

Roles

- GQM Expert (G)
- Facilitator (F)
- Project management (M)
- Software engineer (E)

Level:

- Project (P)
- Organisation (O)
- Not applicable (NA)



APPENDIX 2.10

THE PROFES TEMPLATE FOR: FEEDBACK SESSION REPORT

Table of Contents

1. Introduction
2. GQM Goals Addressed
3. GQM Questions Addressed
4. Conclusions
5. Feedback Session Notes
6. Appendix: Copies of Slides Presented during the Feedback Session

1. INTRODUCTION

Guideline

Include the following information in this section:

Session background	Specify the relevant GQM plan and measurement plan. Specify in which organisational units measures have been collected and analysed.
Session date	Specify when the feedback session has taken place
Session participants	Provide a list of all persons/organisational units that attended the feedback session

2. GQM GOALS ADRESSED:

Guideline

Specify which measurement goals were addressed. Use the following format:

Analyse < **object of study, i.e. process/product** >

in order to < **purpose** >

with respect to < **quality focus** >

from the viewpoint of < **role** >

in the context of <**organisational unit**>

3. GQM QUESTIONS ADRESSED

Guideline

Include here the related GQM questions. They can be found in the GQM plan.

4. CONCLUSIONS

Guideline

Include here a synthesis of the conclusions that resulted from the feedback session.

5. FEEDBACK SESSION NOTES

Guideline

Include here decisions, open items, action points agreed during the feedback session

APPENDIX: FEEDBACK SESSION SLIDES

Guideline

Include a copy of the slides presented during the feedback session



APPENDIX 2.11

THE PROFES TEMPLATE FOR:

TECHNOLOGY APPLICATION REPORT

Guideline

Once technologies from the PPD repository have been used in software projects or improvement programmes, the experience from this technology application should be fed back. Hence, a PPD repository should also contain a growing collection of experience reports from technology applications. These experience reports should be associated with the respective technology definitions. They can become a source of useful information for everybody who is deciding about the application of technologies (e.g., project managers) or who is actually applying them (e.g., software engineers). A recommended structure of such technology application reports is presented in the following.

Technology Application Report	
Technology: <i>(Name of the technology)</i>	...
Project: ... <i>(Name and identifier of the project)</i>	
Product quality goal: ... <i>(What was the product quality goal of the project for which the technology was expected to have particular impact?)</i>	
Process: ... <i>(For which process has the technology been applied in order to contribute to the achievement of the above product quality attribute?)</i>	
Success of technology application: ... <i>(Did the technology application actually help in achieving the required product quality? If yes: Provide some evidence for the product quality achievement. If no: Why did the technology fail?)</i>	
Issues and difficulties encountered during technology application: ... <i>(Were some issues and difficulties encountered when applying the technology in the project? Explain briefly.)</i>	
Recommendations for future applications of the technology: ... <i>(What should be considered and taken care of when applying the technology in future projects?)</i>	
Requests for updates of the PPD repository: ... <i>(Should some information in PPD models that refer to the technology be changed?)</i>	
Author of this technology application report / Contact information: ... <i>(Name, Department, Telephone, e-mail)</i>	
Date: ... <i>(MMM-DD-YY)</i>	



APPENDIX 3

EXAMPLES OF PRODUCT/PROCESS DEPENDENCIES

PROFES has built a repository of product/process dependencies (PPDs) that can be reused by users of the PROFES improvement methodology during the identification of improvement actions. Use of PPDs is described in Step 6 of the PROFES improvement methodology and in Section 7. This appendix provides an overview of the web-based PROFES PPD repository. It contains the following parts:

- Contents of the PROFES PPD repository
- Structure of the PROFES PPD repository

The complete PROFES PPD repository can be accessed via the Internet [PROFES 1999].

Contents of the PROFES PPD Repository

The contents of the PROFES PPD repository are organized in two ways:

- Three types of PPD information: contextual impact PPDs, technological impact PPDs, and process impact PPDs.
- The level of generality of PPD information: specific PPDs for a certain environment, such as a project, company, or business domain, and

generic PPDs that have been generalized from such a specific environment.

This section briefly introduces these views and levels of generality. Afterwards, it surveys example contents of the PROFES PPD repository:

- Generalized process impact PPDs
- Environmentally specific contextual impact PPDs
- Environmentally specific technological impact PPDs
- Environmentally specific process impact PPDs

Three Views on PPD Information

Table 1 illustrates the structure of PPD models. The information contained in a PPD model can be viewed at different levels of abstraction. Each of these views may be more appropriate for certain PPD usage purposes than for others:

Table 1. Example of PPD model

PPD Model	
Product quality	Reliability
Process	Software Architecture Design
Technology	Software Inspections
Context	
Overall time pressure in project	low average high
Experience of project team	low average high
Management commitment to inspections	low high

- (1) The effect of a process on product quality is the highest-level information contained in a PPD model. For example, the software architecture design phase is particularly relevant for ensuring high product reliability. This is called the *process impact view* of a PPD.
- (2) More detailed information is provided when the PPD model also refers to a technology such as software inspection. For instance, it can suggest that software inspections are a good candidate technology for

software architecture design, if high reliability of the software product is to be assured. This is called *technological impact view* of a PPD.

- (3) The most comprehensive information is provided, when the PPD model also contains a context model, as shown above. It describes in which context situations a technology can be expected to have a particular effect on certain product quality. This is also called the *contextual impact view* of a PPD.

More general views on PPDs can be sufficient in certain situations. For instance, process impact PPDs contain all the information necessary for focusing process assessments on those processes that are most critical for certain product quality. In addition, it can be easier to obtain such PPDs, as they require less information. However, the complete, full-fledged PPD models provide the best support for informed decision making in software engineering. The identification of improvement actions, for instance, can benefit from the entire set of information contained in a PPD model.

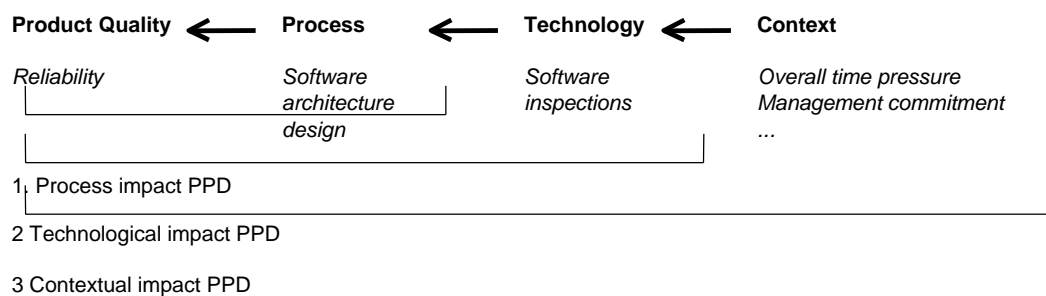


Figure 1: Relations involved in product/process dependencies and the three different views on PPD information

The main concepts contained in a PPD model and the relationship between these concepts are shown in Figure 1. First, software product quality depends on the software engineering processes used for developing the software product. Software reliability, for example, might especially depend on the software requirement analysis process and certain others. Second, the performance of a software engineering process depends on the technologies and practices through which the process is enacted. For example, software inspections can be applied for validating requirements documents. Third, the impact of software engineering technologies on product quality attributes usually depends on certain context factors, such as inspection team members' experience and overall time pressure.

Usually, general PPD information on a higher level of abstraction can in itself be useful for supporting many software engineering tasks and

decisions. Therefore, we recommend a structured form for representing such information using tables, as shown below. A further detailed schema for representing comprehensive PPD information is also introduced.

The Levels of Generality of PPD Information

PPD information can be specific to a certain organizational environment or a business domain, or it can be valid for software engineering in general. When acquiring PPD information and developing PPD models, the results are usually bound to a specific environment. This information can become more generalized by comparing PPD from different environments and formulating their similarities in the form of a new, generic PPD. This strategy of gradual abstraction can be applied to entire PPD repositories, as illustrated schematically in Figure 2. Thus, organization-specific knowledge and experience can be made available to wider audience in the software engineering community.

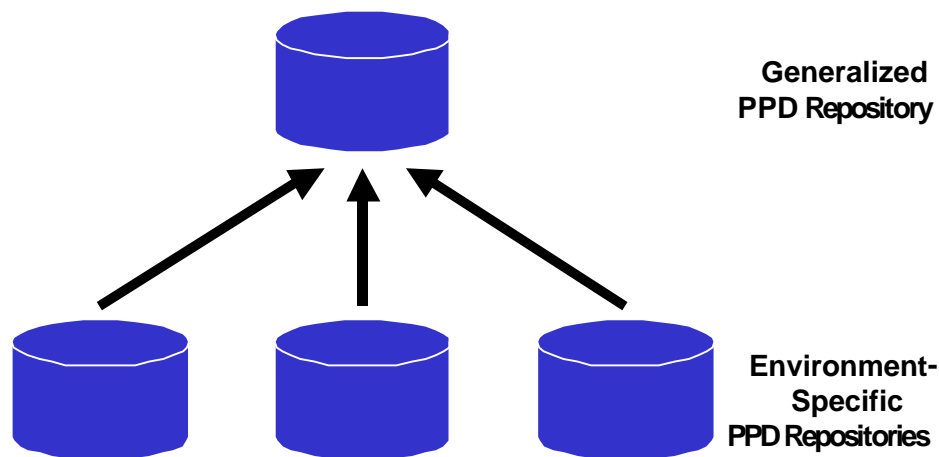


Figure 2: Deriving a generalized PPD repository from environment-specific PPD repositories.

The contents of the PROFES PPD repository reflect the distinction between different levels of generality and different types of PPD information. Figure 3 shows the respective segments of the PROFES PPD repository. It also indicates how information on higher levels of abstraction can be derived from more detailed information. Several examples of different PPD types are presented below.

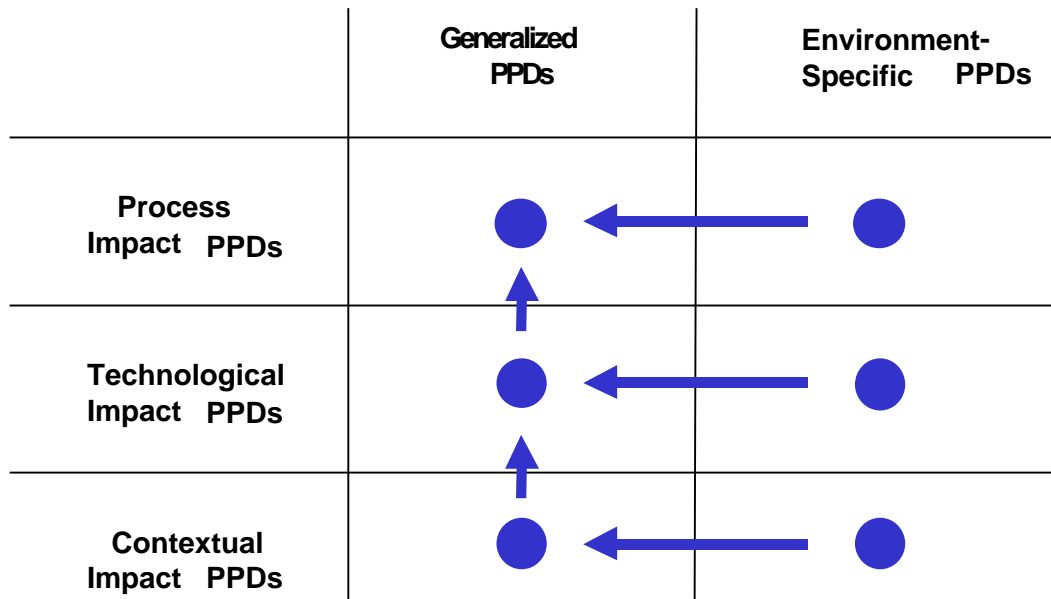


Figure 3: Segments of the PROFES PPD repository and the derivation of generic PPD information into more specialized PPD information

Generalized Process Impact PPDs

The summary of PPD-related findings during PROFES indicates that the following processes are particularly important for achieving high product reliability:

- System Architecture Design
- Software Architecture Design
- Software Detailed Design
- Software Implementation and Testing
- Software Integration and Testing
- System Integration and Testing

Other processes can also be important, but there is less evidence for their general importance in achieving high reliability. Rather, they seem to be complementary or effective only in certain situations. For instance, system requirements analysis seems to be particularly important for product reliability, if new functionality or system features are required that are unfamiliar to the development team. These processes are:

- Lifecycle Methodology
- System Requirements Analysis
- Software Requirements Analysis
- Maintenance
- Configuration Management
- Subcontractor Management
- Reuse

The evidence collected on the importance of individual software engineering processes for reliability is shown in the following table. The left-hand column contains the respective processes. The right-hand column presents statements on evidence and justification concerning the respective sources of the PPD information. The PROFES-A/B/C in right-hand column are different project instances from where the PPD information was gathered. SPICE in right-hand column indicates the SPICE trials [SPICE 1998] as the information source. TRG tags information that have been derived from the Software Engineering Institute's (SEI) C4 Technology Reference Guide [SEI 1997].

Process	Notes
Lifecycle Methodology	<ul style="list-style-type: none"> • PROFES-A: Iterative and incremental lifecycle methodology helps achieve high reliability
System Requirements Analysis	<ul style="list-style-type: none"> • PROFES-A: Requirements reviews, especially for new of functionality, if the underlying hardware is new or if the functionality is developed by a supplier
Software Requirements Analysis	<ul style="list-style-type: none"> • PROFES-A: Requirements reviews, especially for new types of functionality, if the underlying hardware is new or if the functionality is externally developed by a supplier
System Architecture Design	<ul style="list-style-type: none"> • PROFES-A: Design reviews, especially for avoiding late changes and problems due to conflicting requirements • PROFES-C: Place particular emphasis on a thorough architectural design • SPICE: Relative importance wrt. Reliability can be concluded from the identified correlation of good design practices with the ability to satisfy specified requirements (covers Reliability partly)

Software Architecture Design	<ul style="list-style-type: none"> • PROFES-A: Design reviews, especially for avoiding late changes and problems due to conflicting requirements • PROFES-C: Place particular emphasis on a thorough architectural design • SPICE: Relative importance wrt. Reliability can be concluded from the identified correlation of good design practices with the ability to satisfy specified requirements (the latter covers also reliability to some extent)
Software Detailed Design	<ul style="list-style-type: none"> • PROFES-A: Design reviews, especially for avoiding late changes and problems due to conflicting requirements, as well as ensuring sufficiently detailed and correct specifications • TRG: Especially Cleanroom software engineering, personal software process, and software inspections • SPICE: Relative importance wrt. Reliability can be concluded from the identified correlation of good design practices with the ability to satisfy specified requirements (the latter also covers reliability to some extent)
Software Implementation and Testing	<ul style="list-style-type: none"> • PROFES-A: Proper module test process and effective test tools • PROFES-C: Reuse existing and tested code modules • TRG: Especially personal software process
Software Integration and Testing	<ul style="list-style-type: none"> • PROFES-A: Proper integration and system test processes, as well as effective test tools • TRG: Especially Cleanroom software engineering, and personal software process
System Integration and Testing	<ul style="list-style-type: none"> • PROFES-A: Proper integration and system test processes • TRG: Especially Cleanroom software engineering
Maintenance	<ul style="list-style-type: none"> • TRG: Especially Cleanroom software engineering, and personal software process

Configuration Management	<ul style="list-style-type: none">• PROFES-A: Assure consistent use of the correct work products and resolve change conflicts
Subcontractor Management	<ul style="list-style-type: none">• PROFES-A: Select suppliers based on their domain experience, in order to achieve strong implementation of functionality, with particular focus on the testing of the supplied modules
Reuse	<ul style="list-style-type: none">• PROFES-C: Reuse existing and tested code modules

Environment-Specific Contextual Impact PPDs

An example of environment-specific contextual impact is shown in Table 2. It describes how software design inspections have had a particular effect on product reliability within the PROFES-A project. Critical success factors for design inspections in this environment were as follows:

- Size of the inspection team is between 3 and 5 persons
- Experience of the project team is average or high
- Typical complexity of the inspected documents ranges from low to high
- Typical size of the inspected document is small, average, or large
- Overall time pressure in the project is low or average
- Management commitment is high
- Product runs on new hardware platform
- Degree of software developed in-house is greater than 1/3.

The last two context characteristics are particularly interesting: *Product on new hardware platform* and *degree of software developed in-house*. They emphasize the role of design inspections for embedded products and in cases where software is acquired from third parties.

Table 2. Example of PPD model with extended representation scheme

PPD Model	
Product Quality	Reliability
Process	Software Architecture Design
Technology	Software Inspections
Product	Monitoring Software
Viewpoint	Software Engineer
Environment	PROFES-A
Status	Reviewed
Context	
Typical size of inspection teams	1-2 3-5 6-8 9-10
Experience of project team	low average high
Typical complexity of inspected documents	low average high very_high
Typical size of inspected documents	small average large very_large
Overall time pressure in project	low average high
Management commitment for inspections	low high
Product on new hardware platform	new_hw old_hw
Degree of software developed in-house	< 1/3 1/3-2/3 > 2/3

Environment-Specific Technological Impact PPDs







This section presents technological impact PPDs that have been derived from the Software Engineering Institute's (SEI) C4 Technology Reference Guide (TRG) [SEI 1997] [Birk and Kiesgen 1999]. The TRG is a collection and classification of software technologies. Its purpose is to foster technology dissemination and transfer. Each technology is classified according to those processes to which it can be applied (application taxonomy) and according to qualities those software systems qualities that can be expected as a result of applying the technology (quality measures taxonomy). These classifications have been comprehensively reviewed by a large number of noted software engineering experts. This accumulated expert opinion can be used as a source of evidence for the impact of software engineering processes on overall process performance. Table 3 shows the technological impact PPDs contained in the TRG that refer to the software design process.

Table 3. Technological impact PPDs contained in the TRG that refer to the software design process

Process	Quality Measures (Process Performance Attributes)											Technology
	Correctness	Verifiability	Availability/ Robustness	Reliability	Efficiency/ Resource utilization	Usability	Maintainability	Understandability	Interoperability	Reusability	Cost of ownership	
Architectural design	X							X		X		Architecture description language
	X							X		X		Module interconnection language
Database design		X			X		X	X	X			Graphic tools for legacy data base migration
							X					Object-oriented databases
Interface design							X		X			COTS and open systems
						X	X					Graphical user interface builders
Select or develop algorithms	X				X							Algorithm formalization
							X					Component-based software development / COTS integration
Detailed design	X		X	X			X	X				Cleanroom software engineering
	X			X								Hybrid automata
							X			X		Object-oriented design
			X	X							X	Personal software process for module-level development
				X	X		X					Rate monotonic analysis
	X		X	X			X					Software inspections
System analysis and optimization				X	X		X					Rate monotonic analysis

Environment-Specific Process Impact PPDs

An analysis of the SPICE database [SPICE 1998] was reported in [Emam and Birk 1999]. The objective was to validate the predictive measures of ISO 15504 process assessments. From this analysis, the following PPD-related information on the project performance impact of selected ISO 15504 processes could be identified. Project performance as defined in the SPICE study is related in some respects to the product quality notion as used in PROFES. For this reason, the SPICE study provides baseline information about process impact PPDs that can be used within the PROFES PPD repository. Examples of such process impact PPDs are shown in Figure 4.

	<i>Product-related project characteristics</i>	Ability to meet budget commitments	Ability to meet schedule commitments	Ability to satisfy specified requirements	Staff productivity
Processes	Develop SW requirements				
	Develop SW design				
	Implement SW design				
	Integrate and test SW				

(El Emam, Birk, 1999; based on the SPICE trials) *Large SW organisation: IT staff ^a 50 persons*

Figure 4: Impact relationships for large SW organizations with IT staff of more than 50 people

Structure of the PROFES PPD Repository

The core part of the PROFES PPD repository is the collection of PPD models. These should be embedded in (1) a set of relevant index structures or taxonomies, and (2) a set of definitions or glossaries that assure the clear understanding of the information contained in the PPD models. There may also be a collection of case reports that document experience with PPDs and software engineering technology made during previous projects within the organization.

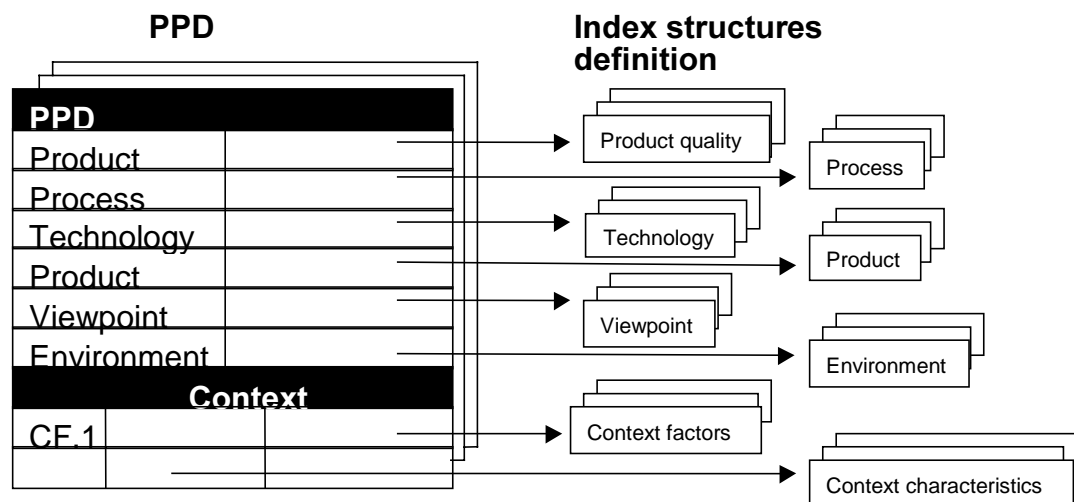


Figure 5: Overview of PPD repository structure.

The two basic taxonomies that should form the index structure of a PPD repository are: (1) a taxonomy of product qualities, and (2) a taxonomy of software engineering processes

The PROFES PPD repository uses a taxonomy of product qualities that is based on the ISO/IEC 9126 standard of software product quality attributes. In addition, “product qualities” software development cost and time to market are included. These are relevant improvement goals for which a company might be interested in finding the relevant processes and appropriate technologies. Just like other product qualities such as reliability, functionality, and maintainability, they can be attributed to the final software product and the perception that customers have of the product, that is the cost of developing the product and the time needed to access the market. Since any fixed taxonomy is to some extent restricted and although not every possible category necessary in the future might be foreseen, we also recommend including a generic category called “other”.

The product quality taxonomy used in the PROFES PPD repository is shown in Table 4.

Table 4. The PROFES PPD Repository's product quality taxonomy

Functionality	Portability
Reliability	Software Development Cost
Usability	Time to market
Efficiency	Other
Maintainability	

The process taxonomy of the PROFES PPD repository is based on the BOOTSTRAP V3.0 process dimension shown in Table 5. The original collection of the BOOTSTRAP V3.0 processes is quite comprehensive and detailed. However, for serving as an index structure for PPD repositories, it should be supplied with the following additional categories: lifecycle methodology, measurement, process assessment, and other.

There are also other taxonomies that can ease the use of a PPD repository and maintenance, but are not essential, such as: (1) a taxonomy of viewpoints, (2) a taxonomy of PPD model status, and (3) a taxonomy of environments. Appropriate viewpoints are the three basic software engineering roles of software engineer, quality assurance, and management. They document from whose viewpoint the PPD information has been acquired. Recommended status categories are: Initial, reviewed, and validated empirically. These indicate different degrees of confidence one might have in a PPD model. The environment taxonomy should reflect the structure of the software organization from which the PPD information has been acquired and for which it will be used. Examples of categories are project or department names, as well as technological domains and branches of software engineering, e.g. embedded systems, information systems, or client/server development.

Table 5. The PROFES PPD Repository's process taxonomy

ORG.1	Business engineering	MAN.1	Project management
ORG.2	Product strategy formulation	MAN.2	Quality management
ORG.3	Human resource management	MAN.3	Risk management
ORG.4	Infrastructure management	MAN.4	Subcontractor management
ORG.5	Reuse	MAN.5	Product management
PRO.1	Process definition	SUP.1	Documentation
PRO.2	Measurement	SUP.2	Configuration management
PRO.3	Process control	SUP.3	Quality assurance
PRO.4	Process improvement	SUP.4	Verification
		SUP.5	Validations
PLC.1	Product requirements specification	SUP.6	Joint review
PLC.2	Product design	SUP.7	Audit
PLC.3	Systems design and implementation	SUP.8	Problem resolution
PLC.4	Systems integration and testing	CUS.1	Acquisition
PLC.5	Production and installation	CUS.2	Customer needs management
		CUS.3	Supply
		CUS.4	Operation
		CUS.5	Customer support
ENG.1	System requirements analysis	CUS.6	Product control and promotion
ENG.2	System architecture design	CUS.7	Production ramp-up
ENG.3	Software requirements analysis	TEC.1	Technology innovation
ENG.4	Software architecture design	TEC.2	Technology support
ENG.5	Detailed software design	TEC.3	Technology for software life-cycle support for I-c independent
ENG.6	Software implementation and testing	TEC.4	Tool integration
ENG.7	Software integration and testing		
ENG.8	System integration and testing		Lifecycle methodology
ENG.9	Maintenance		Measurement
ENG.10	Migration		Process assessment
ENG.11	Retirement		Other

Every term used in a PPD repository taxonomy should be supplied with a clear and unambiguous definition. These definitions can be provided in the form of a glossary or data dictionary. Examples of product quality definitions used in the PROFES PPD repository:

- Functionality is defined as “the capability of the software to provide functions that meet stated and implied needs when the software is used under specific conditions”. The concept of functionality can be further divided into suitability, accuracy, interoperability, and security
- Reliability is defined as “the capability of the software to maintain the level of performance of the system when used under specific conditions”. Reliability can be further categorized into the sub-characteristics of maturity, fault tolerance, and recoverability
- Maintainability is defined as “the capability of the software to be modified”. Modifications may include corrections, improvements, or adaptation of the software to environmental changes, and in requirements and functional specifications. Maintainability can be further categorized into analysability, changeability, stability, and testability.

Likewise, processes are defined as follows:

- Software requirements analysis: The purpose of the software requirements analysis process is to establish, analyze, and refine the requirements of the software item of the system.
- Software detailed design: The purpose of the software detailed design process is to establish a software detailed design that effectively accommodates the software requirements and refines the major software components into lower level software units which can be coded, compiled, and tested.
- Software integration and testing: The purpose of the software integration and testing process is to integrate the software units with each other producing software that will satisfy the software requirements.

The definitions of software engineering technologies are particularly important contents of a PPD repository. PPD repositories are typically used for identifying improvement actions. Usually, these improvement actions are technologies that are newly introduced to a software project. Examples are advanced inspection or testing methods, techniques for requirement documentation, or CASE tools. In this case, a technological definition should be as operational as possible. It might be a method process description or a comprehensive user manual for a new notation. These definitions can be supplied with additional documentation and

training that facilitate the introduction and application of the new technology.

It can be quite effort consuming to develop a comprehensive and operational technology definition. For some technologies, generic definitions exist in the public domain that can be adapted to the specific needs and characteristics of a given software organization. Such a technological definition for software inspections is documented in [Birk and Kiesgen 1999]. In other cases, it may be sufficient to rely for some time on short textual definitions of the technologies, and supply them with references to literature for further information. When such a technology needs to be adapted for a software organization, it becomes worthwhile to provide a more comprehensive and operational technological definition as part of the PPD repository. Examples of textual technological definitions contained in the PROFES PPD repository are:

- **Cleanroom software engineering:** Cleanroom software engineering is an engineering and managerial process for the development of high-quality software with certified reliability. Cleanroom was originally developed by Dr. Harlan Mills. The name “Cleanroom” was taken from the electronics industry, where a physical clean room exists to prevent introduction of defects during hardware fabrication. Cleanroom software engineering reflects the same emphasis on defect prevention rather than defect removal, as well as certification of reliability for the intended environment of use
- **Personal Software Process for Module-Level Development:** Personal Software Process (PSP) is a framework of advanced process and quality techniques to help software engineers improve their performance and that of their organizations through a step-by-step, disciplined approach to measuring and analyzing their work. Software engineers that use the PSP can substantially improve their ability to estimate and plan their work and significantly improve the quality, i.e., reduce the defects, in the code they develop. PSP is a result of research by Watts Humphrey into applying process principles to the work of individual software engineers and small software teams. The objective was to transfer the quality concepts of the Capability Maturity Model (CMM) for Software to the individual and small team level.
- **Software Inspections:** Software inspections are a disciplined engineering practice for detecting and correcting defects in software artifacts and preventing their leakage into field operations. Software inspections were introduced at IBM in the 1970s by Michael Fagan, who pioneered their early adoption and later evolution. By detecting defects early and preventing their leakage into subsequent activities, the higher cost of later detection and rework is eliminated; this is essential for

reduced cycle time and lower cost. Software inspections provide value in improving reliability, availability, and maintainability.

The structure of the PROFES PPD repository can serve as a blueprint for organizing customized, organization-specific PPD repositories or repositories of good software engineering practice.

References

[PROFES 1999] PROFES repository of product/process dependencies. <http://www.iese.fhg.de/Profes>.

[Birk and Kiesgen 1999] Andreas Birk and Thomas Kiesgen. Identification of product/process dependencies from the SEI C4 technology reference guide. Fraunhofer IESE Technical Report, Kaiserslautern, Germany, 1999.

[Emam and Birk 1999] Khaled El Emam and Andreas Birk. Validating the ISO/IEC 15504 measure of software requirements analysis process capability. Fraunhofer IESE Technical Report IESE-003.99. Kaiserslautern, Germany, 1999.

[SPICE 1998] SPICE Project. "Phase 2: Trials", interim report. 1998.

[SEI 1997] Software Engineering Institute: C4 Software Technology Reference Guide – A Prototype. Handbook CMU/SEI-97-HB-001, Software Engineering Institute, 1997.



APPENDIX 4

COST/BENEFIT MODELS OF PROFES

PROFES has analysed the cost and benefit of applying the PROFES improvement methodology. As the result of this work, various benefits from product-focused process improvement have been identified and a collection of effort models has been built. These results are documented in the web-based PROFES cost/benefit repository [PROFES 1999].

This appendix describes the contents of the PROFES cost/benefit repository, outlines a typical PROFES improvement programme and the effort involved, and presents two selected cost models that describe the effort of applying elements of the PROFES improvement methodology.

Contents of the PROFES Cost/Benefit Repository

The web-based PROFES cost/benefit repository summarizes the experience gained from applying the PROFES improvement methodology in three industrial application projects. The contents are as follows:

- Benefit from product-focused process improvement
- Effort models of product-focused process improvement
- Cost/benefit ratio of product-focused process improvement
- Validation of the PROFES improvement methodology
- Guidelines and lessons learnt

Most of the cost/benefit information contained in the repository is based on qualitative evidence. This means that the results are derived from observa-

tions, survey questions, and statements of participants of the software projects and improvement programmes. Where possible, additional quantitative evidence is provided based on measurement data from the PROFES application projects.

An Effort Scenario of Running a PROFES Improvement Programme

This section presents the scenario of a typical PROFES improvement programme and demonstrates that the application only requires moderate effort. Both the design of the scenario and the effort figures are based on experience from the three industrial pilot applications of PROFES.

A Typical PROFES Improvement Programme

The course of a typical PROFES improvement programme is shown in Figure 1. It involves five main phases, some of which are carried out concurrently. The first phase covers the start up and goal setting. It establishes the infrastructure necessary for applying the PROFES improvement methodology, obtains the necessary management commitment and sponsorship, and identifies the product quality goals to be achieved through the improvement programme.

The product quality goals can be used for setting up a subsequent process assessment and a measurement programme that represent two additional phases of the improvement programme. The defined goals help to focus the process assessment and the measurement programme on those areas of the software organization and its processes that are important for the achievement of the required product quality. This focusing is supported by the identification of product/process dependencies (PPDs).

The identification and deployment of PPDs is an additional phase of the improvement programme that can already begin during the initial goal setting phase. It is usually followed by another phase, namely the early implementation of improvement actions. Additional improvement actions might be suggested by the process assessment. The measurement programme allows for monitoring and controlling the success of the improvement actions. Its results can lead to the definition of additional or updated product quality goals, and can suggest further improvement actions. This usually leads to a new improvement programme cycle, which again follows the five phases.

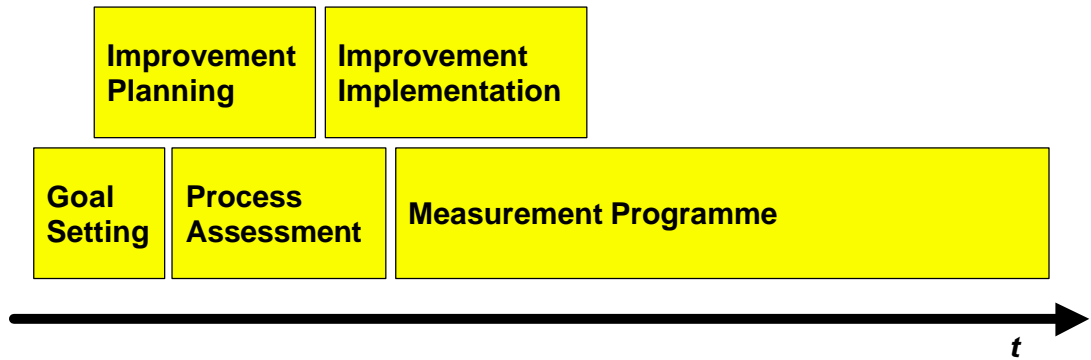


Figure 1: Outline of a typical PROFES improvement programme.

Duration and Effort of a Typical PROFES Improvement Programme

The typical duration of one cycle in the PROFES improvement programme is less than a year. At the end of this period, the first substantial process and product quality improvements can usually be seen. The total effort for a typical PROFES improvement programme for a project with ten software engineers is about 6.5 person-months. Most of the effort is made by the process assessment team and the measurement programme manager. The involvement of management and the software development team is particularly low. It takes place in a focused manner only at key events in the improvement programme. Hence, the PROFES improvement method also scales up well for much larger teams but only requires slight additional effort.

Table 1: Effort and duration of a typical PROFES improvement programme.

Phase of the improvement programme	Effort (person-months)	Time (calendar weeks)
Start up & goal setting	0.5	2
Process assessment	2.5	6
Measurement programme	2.5	40
Identification of PPDs	0.5	2
Improvement implementation	0.5	2
Total	6.5	52

Table 1 shows the typical effort and duration of each phase in a PROFES improvement programme. This effort is stated in person-months. Throughout this section, one person-month equals four person-weeks, twenty person-days, and 160 person-hours. In the following, assumptions underlying the calculation of effort figures are explained.

- *Start up & goal setting*

- The *roles* involved in this phase are the improvement programme manager, i.e. usually a process or quality engineer. Below, this role is also referred to as *IP manager*, the manager of the project in which the improvement programme is performed, and two senior engineers from the project team.
- The *activities* conducted in this phase are a two-hour start-up meeting, an introduction to the project for the improvement programme manager, a one-day workshop for planning and goal setting, and general communication and preparation activities.
- The *effort* per activity expected from each participating role is shown in the following table (effort in person-hours):

	Manager	Software engineer	IP manager	Total
Start-up meeting	2	4	2	8
Project introduction		4	2	6
Goal-setting workshop	8	16	8	32
Communication & preparation	2	4	28	34
Total	12	28	40	80

- *Process assessment*

Process assessment in the example scenario is the same as the example effort model for BOOTSTRAP process assessments described in the following section. Its total effort is about 400 person-hours, which equals 2.5 person-months.

- *Measurement programme*

The measurement programme within the example scenario is the same as the example effort model for GQM measurement programmes described below. Only the duration of the measurement programme in the scenario is longer than the one underlying the example effort model. For this reason, some further effort must be added:

- The baseline model involves only one cycle of data collection and feedback session, which lasts two months. In the example case, there are four additional such cycles.
- Each two-month period of data collection requires 13 hours of data collection effort by the engineers, i.e. 10 engineers for 8 weeks of data collection and 10 minutes effort per week). The GQM expert needs four hours for data validation. This adds up to 17 hours in all. Four such two-month periods require an additional 68 hours.
- Each additional feedback session lasts two hours. Participants are the project manager, seven out of ten software engineers, and the GQM expert. The GQM expert needs an additional 16 hours for preparation of the feedback session and for follow-up actions. This adds up to 34 hours in total (manager, 2 hours; all software engineers, 14 hours; GQM expert, 18 hours). The total effort for four additional feedback sessions is 136 hours.

- Hence, the total effort for the GQM measurement programme within the PROFES improvement programme is 385 person-hours. This consists of 181 person-hours for definition of the measurement programme and the first measurement cycle, 68 person-hours for four additional data collection cycles, 136 person-hours for four additional feedback sessions, making a total of about 2.5 person-months in all.
- *Identification of PPDs*
 - The *roles* involved in this phase are the improvement programme manager who carries out PPD identification and modelling work, the manager of the project in which the improvement programme is performed, and two senior engineers from the project team.
 - The *activities* conducted in this phase are preparation by the IP manager, interviews of senior engineers or management for acquiring or validating PPD models, and a final review and approval phase. The IP manager sets up an initial collection of PPD models based on an existing PPD repository, literature, and past project data. These initial PPDs are extended and validated through interviews. The collection of PPD models is then reworked by the IP manager and reviewed and approved by the project manager and the engineers.
 - The *effort* per activity expected for each participating role is shown in the following table (effort in person-hours):

	Manager	Software engineer	IP manager	Total
Preparation			32	32
PPD Interviews	2	8	24	34
Review & approval	2	4	8	14
Total	4	12	64	80

- *Improvement implementation*
 - The effort for improvement implementation depends very much on the actual improvement actions. In this example scenario, we assume that the introduction of systematic software inspections is the improvement action. The team is already used to informal, ad-hoc code reviews.
 - The *roles* involved in the introduction of software inspections are the entire project team, i.e. all ten engineers, the improvement

programme manager, and an external technology expert in charge of training and consulting for software inspections.

- The *activities* conducted in this phase assume that software inspections are introduced on the job. They are begin by adapting the software inspections process and tools to meet project requirements. A technical briefing and short training session is then held, and the technology expert supervises the first series of inspection meetings.
- The expected *effort* per activity for each participating role is shown in the following table (effort in person-hours):

	Software engineer	IP manager	Expert	Total
Preparation & adaptation		4	8	12
Technical briefing	40	4	4	48
Training			20	20
Total	40	8	32	80

Effort Model Example of the PROFES Improvement Methodology

The PROFES cost/benefit repository contains various models that help to predict and evaluate the impact of product-focused improvement programmes. The main areas to be affected are software organizations, projects, and the developed software products. Each cost/benefit model must be viewed in the context of certain associated assumptions, such as the size of the software organization, its experience of process improvement initiatives, other ongoing improvement activities, etc. This contextual information is provided in detail with every cost/benefit model contained in the repository. In this section, we illustrate the characteristics and use of cost/benefit models by presenting two selected examples of effort models for applying elements of the PROFES improvement methodology. The first model addresses the effort of BOOTSTRAP process assessments. The second model describes the effort of GQM measurement programmes.

Depending on the assessment method chosen, additional licensing costs must be expected. Furthermore, assessment and measurement activities can be performed by external consultants, and specific training for internal personnel can become necessary. In these cases, additional costs might

occur. Basic GQM training effort, which is a standard part of every measurement programme, is included in the PROFES effort models.

Effort for BOOTSTRAP Process Assessments in Organizations of Average Complexity

This section presents an operational effort model of BOOTSTRAP process assessments that is based on experience of PROFES applications. The model can be used to assist planning and the estimation of effort in BOOTSTRAP process assessments. It is operational in the sense that it includes information that allows the model to be adapted to suit varying characteristics and constraints present in the organizational environment where the process assessment will be performed.

One of the results of applying PROFES was that the effort necessary for BOOTSTRAP assessments can vary considerably depending on the degree of complexity of the organizational structure in the assessed organization and projects. Therefore, four different effort models have been defined for BOOTSTRAP assessments: Full assessment and focused re-assessment for each organization with either relatively complex organizational structures or highly complex organizational structures.

In the following, the effort model for organizations of average complexity is described in terms of a scenario with specific assumptions and their conclusions. First, the assumptions that underlie the scenario are stated. Second, prototypical effort figures are provided for each process step and role involved in the process. Third, the cost structure is explained.

Assumptions

The following assumptions have been made when setting up the model:

- 1 lead assessor
- 1 assessor
- 3 SPU interviewees
- 2 managers
- 1 software engineer
- 2 projects with 3 project interviewees each
- 1 manager for each of the two projects

- 2 software engineers for each of the two projects
- 1 facilitator

Effort Model

The effort model is shown in Table 2. Effort is stated in person-hours per activity in the BOOTSTRAP process assessment, and per role involved in the assessment. The activities are listed in the left-hand column. The roles are shown in the top row of the table.

Table 2. An assessment effort model.

Activity	Lead Assessor	Assessor	Engineer	Manager	Facilitator	Total
Preparation	24	20		1	16	61
Opening briefing	7	7	15	12	1	42
Global site assessment	6	6	2	4	3	21
Project assessment	15	15	10	5	6	51
Evaluation	40	40				80
Review			5	4		9
Final meeting	1.5	1.5	7.5	6		16.5
Report preparation	80			2	16	98
Report review			10	8		18
Total	173.5	89.5	49.5	42	42	396.5

Cost Structure

- Preparation:

For preparing the assessments, both assessors each need half a week. Additionally, the lead assessor needs half a day for inviting the participants, etc. The manager role and the facilitator are also involved, with an effort of one hour or two days respectively, for planning and providing information necessary for the preparation of the assessment.

- Opening Briefing:

The briefing requires an effort of three hours per participant. The assessors need four hours more to prepare the briefing. The facilitator works one hour to prepare the opening briefing and send out invitations.

- Global Site Assessment:

Interviews lasts for two hours each, with participation by the lead assessor, the assessor, and one interviewee. The

facilitator is only partly involved in the interviews, for example by inviting the participants and making project documentation available, which results in an average effort of one hour per interview. This also applies to the facilitator's project assessment effort.

- Project Assessment:

The six interviews last 2,5 hours each, with participation by the lead assessor, the assessor, and one interviewee. The facilitator is involved in the same way as during the global site assessment interviews.

- Evaluation:

Interview data evaluation requires about one week per assessor. Usually, the evaluation is scheduled to take place on-site at the assessed organization immediately after each assessment interview.

- Review:

Reviews are carried out by the interviewees, and each lasts about an hour.

- Final Meeting:

The final meeting lasts 1.5 hours, attended by the assessors and the interviewees.

- Report Preparation:

Writing the report is mainly the work of the lead assessor, requiring an effort of about two weeks. In addition, the manager is involved with a two-hour effort, and the facilitator participates for two days.

- Report Review:

The report is reviewed by the SPU and project interviewees, requiring about two hours per person, similar to the first review of the assessment results.

Duration and Effort of GQM Measurement Programmes for Organizations with Experience in Applying GQM

This section describes the effort necessary for running a GQM measurement programme in an organization with experience in GQM-based measurement. Like the previous model on BOOTSTRAP assessments, it is based on the experiences of the PROFES applications. The structure of the scenario is also similar: assumptions and prototypes of effort figures and cost structures. In addition, several variations of effort figure prototypes were identified.

Three different effort models for GQM measurement have been identified in all: Routine application of GQM measurement, transition to GQM measurement, and introduction of GQM measurement in organizations with little measurement experience.

Underlying Assumptions

The following assumptions were made in this model:

- Participants:
 - 1 GQM expert
 - 1 manager as interviewee
 - 4 non-management roles as interviewees (3 software engineers and 1 software quality engineer)
- GQM goals:
 - 3 GQM goals with the same object, quality focus, purpose, and environment; and with 3 different viewpoints, i.e. management, quality assurance, and software engineer.
- Size of project team:
 - The project team consists of 10 software engineers
- Support infrastructure:
 - There is an established tool infrastructure for supporting the GQM measurement programme with computer tools and paper-based tools. It involves text processing systems with appropriate templates for measurement planning, a database system containing the measurement database, on-line data collection support connected to the measurement database, and utilities for generating reports and presentation material from the measurement database.

- The available support tools are applied by the roles described above.
- Training and briefings:
 - There is no special need for training because the measurement programme participants are used to GQM measurement
 - There is no need for extensive briefings. The primary role of briefings is to give start signals and to provide continual support and motivation to the team members.

Effort Model

Table 3 illustrates the effort model. The effort is stated in person-hours per activity in the GQM measurement programme and per role involved. The activities are listed in the left-hand column. The roles are shown in the top row of the table.

Table 3. An effort model for GQM measurement programme.

Activity	GQM Expert	Manager	Software Engineer	Total
Prepare measurement programme	4	2		6
Identify and define GQM goals	8	1	4	13
Prepare and conduct GQM Interviews	24	1.5	6	31.5
Develop GQM plan	32		2	34
Develop measurement plan	8		1	9
Perform data collection	8	0.5	18	26.5
Perform data analysis and interpretation	32	2	14	48
Package experience	8	1	4	13
Total	124	8	49	181

Cost Structure

According to the activities performed in a GQM measurement programme and the roles involved, the following cost structure is typical for the routine application of GQM:

- Prepare measurement programme:

The GQM expert makes preparations with limited involvement by the manager. The effort is 4 hours for the GQM expert and 2 hours for the manager.

- Identify and define GQM goals:

The GQM expert does most of the work. Since there is one GQM goal with three viewpoints, this leads to an effort of one person-day for the preparation work, communication with the measurement programme sponsor (i.e. the manager), and evaluation of the appropriateness and feasibility of the GQM goal. The manager and the software engineers mainly contribute to the selection of improvement goals and GQM goals, which requires about one hour per person.

- Prepare and conduct GQM interviews:

No opening briefing sessions are held. However, all interviewees are separately briefed at the beginning of each interview. A GQM goal interview lasts about 1.5 hours per interviewee. The time required for one interviewee is divided into one hour for the actual interview and half an hour for any further clarification after the actual interview.

All the interviews are conducted by the GQM expert (5 x 1.5 hours). Approximately two extra days are needed for preparing the interviews and for revising the results.

- Develop GQM plan:

The GQM expert usually performs this activity, which requires an overall effort of four days. The software engineers are involved with 2 hours of effort in total for reviewing and clarification.

- Develop measurement plan:

As in the previous phase of the measurement programme, this activity mainly involves the GQM expert, and requires one day's work. The engineers need an hour to clarify issues and briefly review the data collection plan.

The cost data for the three remaining phases apply for one data collection cycle that lasts two months. It is concluded by a feedback-session.

- Perform data collection:

An opening briefing is held with an effort of half an hour per participant, i.e. the expert, management, and all the project participants.

Every data collector needs to spend about ten minutes per week on the task. The GQM expert's work can be split into half a day for preparing and conducting the briefing, and another half-day for data validation, total one day. The manager spends half an hour and the software engineer 18 hours in all.

- Perform data analysis and interpretation:

The GQM expert needs four days for preparing and conducting the feedback-session, which lasts two hours. Other participants in the feedback-session are the expert, the manager, and seven out of ten software engineers.

- Package experience:

The expert packages experience from the measurement programme. He or she prepares and conducts the package review, which requires one day. The manager and the four software engineers (who participated as interviewees during the GQM interviews) work one hour per person by reviewing and approving the packages.

Variations

There are various cost-linked factors that can affect GQM measurement work. These variants are listed below and their impact is explained:

- Number of GQM goals:

If the number of GQM goals increases, then the amount of work also increases in all activities starting with the "Identify and Define GQM goals" step, as the usefulness of the goals has to be discussed and a larger number of measures probably need to be identified. Due to the fact that there are more measures, more effort is also needed to develop the measurement plan. In addition, more time is spent on data collection. The analysis and packaging effort will also increase with every additional GQM goal.

- Size of project team:

- Number of project participants

If the number of project team members increases, this will necessitate a greater number of GQM interviews in order to involve a sufficiently large number of engineers. The total effort for data collection will also increase. Feedback sessions should possibly be split into more than one meeting with different

participants, in order to retain an optimal number of participants for effective discussions.

- Geographical distribution of project:

If a project is distributed among geographically dispersed sites, then more effort will be necessary to co-ordinate and manage the measurement programme. This mainly increases the GQM expert's workload. Repeated feedback sessions may possibly be held at different sites, and their results must be summarized.

- Experience with tools/forms:

If there is not sufficient experience of using data collection forms or other tools in the measurement programme, then the total effort can be expected to increase. This is due to the fact that familiarity with measurement programmes and the tools used are the main reasons for a low total effort for the routine application of GQM measurement.

References to Appendix 4

[PROFES 1999] PROFES 1999, "*PROFES PPD repository*."
<http://www.iese.fhg.de/Profes>.



APPENDIX 5

SPECIFIC TOOLS USED DURING THE PROFES PROJECT

The purpose of this section is to introduce the specific tools used during the PROFES project as follows:

- **Bootsample** – a process assessment tool
- **GQMaspect** – a tool for building GQM plans
- **MetriFlame** – a tool for managing measurement plans and data
- **Trend Analysis** – a tool for monitoring process capability

Bootsample – How to Use Tool Support in Process Assessment

Process assessment is used to provide a solid foundation for process improvement programmes. Based on the assessment findings, current process capabilities are identified and then appropriate improvement recommendations are defined. However, collecting and analysing the assessment data manually is a very tedious task. Therefore, tool support is necessary to perform assessment efficiently.

In this chapter, we will introduce an assessment tool called Bootsample. The tool was used in the PROFES project to support process assessments. During the PROFES project, Bootsample was also enhanced to support assessment of product-based embedded systems development.

Introduction

Bootsample is an assessment tool that has been developed by the BOOTSTRAP Institute for use as a support tool by assessors during Bootstrap 3.0 evaluations. Bootstrap 3.0 is an ISO 15504¹-compliant process assessment and improvement methodology. Bootsample can be used for defining assessment context and scope, collecting findings, and calculating assessment profiles and capability ratings.

Bootsample fully supports the Bootstrap 3.0 architecture, including all the defined processes and capability levels. In addition, the tool also includes the new embedded specific processes that were defined during the PROFES project. Bootsample can be used to support those PROFES steps that involve process assessment activities. These steps are as follows: Determine current process capability (Step 4), Implement and monitor improvements (Step 10) and Evaluate results (Step 11).

Main Functions

Table 1 describes the most important functions of Bootsample for each assessment subtask. These functions are further discussed below:

¹ ISO 15504 is also known as SPICE. BOOTSTRAP was the assessment method used during the PROFES project.

Table 1. Supporting functionality offered by Bootsample for each assessment subtask

Assessment subtask	Support offered by Bootsample
Preparation	<ul style="list-style-type: none"> • Management of information related to assessment, target organization, SPU, and project • Classification of assessed organization, SPU, and project for the purpose of benchmarking
Implementation	<ul style="list-style-type: none"> • Management of findings and scores
Improvement planning	<ul style="list-style-type: none"> • Calculation and printing of capability profiles • Printing of BOOTSTRAP and SPICE reports • Analysis of process trends

At first, a new assessment entry is created in the Bootsample database by entering name, date, place, names of assessors, etc. We recommend that the database be a part of the PROFES experience base. Information on the organization to be assessed is stored in the database. This information both defines the organization (name, address etc.) and classifies it (branch of industry, number of employees, etc.) In the same way, information regarding either the target software producing unit (SPU) or target project(s) is collected and stored. All the work explained so far is usually done during the preparation phase. The organization context definition window of the Bootsample tool is shown in Figure 1:

Figure 1. The Bootsample organization definition window

In the execution phase the actual evaluation of the SPU or the project is done. This includes both recording findings and scoring practices. With Bootsample, assessors can easily return to their previous notes and

evaluations to complete them in the light of newly collected information. The Bootsample scoring window is shown in Figure 2. Evaluation includes functions related to printing SPICE and BOOTSTRAP reports, and the results of the assessment.

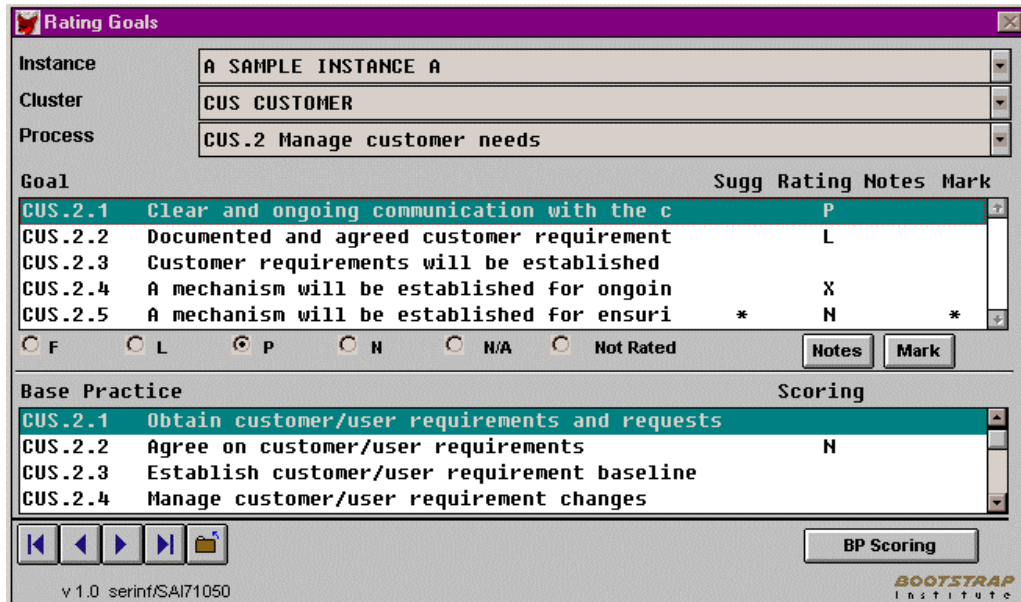


Figure 2. Bootsample scoring window

GQMaspect – How to Use Tool Support in GQM Planning?

GQMaspect (**G**QM **a**bstraction **s**heet and **G**QM **p**lan editing and construction **t**ool) is a research prototype by Fraunhofer IESE that supports the planning phase of GQM-based measurement programmes [Basili et al. 1994b]; [Briand et al. 1996]. The tool has been developed together with the University of Kaiserslautern within the context of several research and industry projects aiming to introduce and conduct measurement-based software process improvement programmes, among others the CEC-funded projects CEMP (ESSI project number 10358 [Latum et al. 1998]), PERFECT (ESPRIT project number 9090 [Birk et al. 1998]), and PROFES.

The functionality of *GQMaspect* focuses on the definition and maintenance of GQM plans. The tool provides the following support [Hoffman et al. 1996]:

- Templates for the definition of GQM goals
- Templates for the definition of GQM abstraction sheets and GQM plans
- Editing functions for constructing GQM abstraction sheets and GQM plans, involving the reuse of already existing GQM documentation
- Automatic generation of GQM plans from GQM abstraction sheets and vice-versa.

Since *GQMaspect* has been written in the programming language 'Java', it runs on all major platforms such as Sun Solaris, Macintosh OS, Windows 9X, and Windows NT.

A brief overview of the most important functions of *GQMaspect* is given in Figure 3. It shows how the tool is integrated into the GQM process. The GQM process is sub-divided into phases such as 'Identify GQM goals', 'Produce GQM plan', 'Produce measurement plan' and 'Collect and validate data'. Detailed descriptions of these steps are provided in [Gresse et al. 1995]. *GQMaspect* supports the phase 'Produce GQM plan' which is an iterative process. The GQM goals will have already been defined when this phase is entered. In the first step, the GQM expert interviews those individuals who are specified in the 'viewpoint' aspect of a GQM goal. The results of these interviews can be structured using abstraction sheets, one for each interview. All necessary abstraction sheet templates and features for editing them are available in *GQMaspect* (cf. Figure 4). The next step is to merge the set of abstraction sheets into one 'summarized' abstraction sheet.

Usually, the first summarized abstraction sheet will still contain several inconsistencies and sometimes information is missing. In order to resolve the inconsistencies and provide the missing information, follow-up interviews must be held. When the summarized abstraction sheet is consistent and complete, a GQM plan can be generated from the summarized abstraction sheet using *GQMAspect*. Any additional information not included in the GQM abstraction sheet but which is necessary for performing GQM-based measurement programmes is entered into the GQM plan.

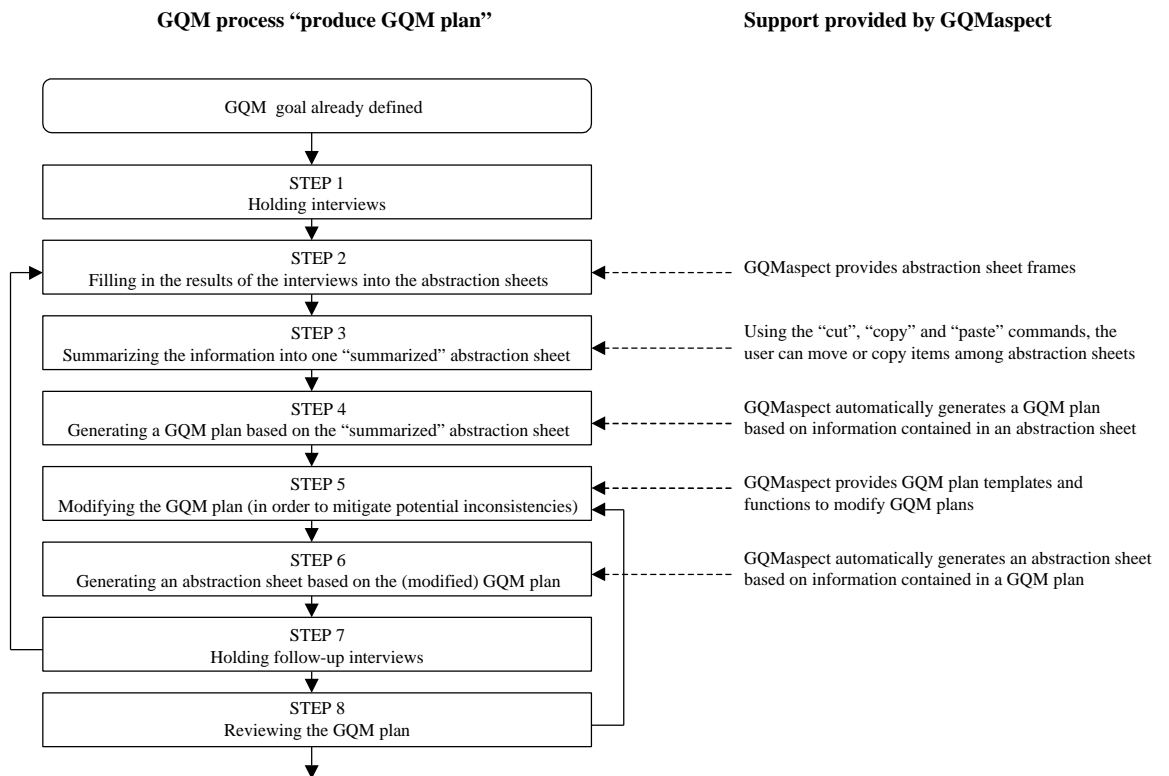


Figure 3. An overview of the GQM process ‘Produce GQM plan’ and the role of GQMAspect in this process.

During the construction of the GQM plan, a need for additional information may arise which can make new interviews necessary. For that purpose, *GQMAspect* offers the reverse generation of GQM abstraction sheets from existing GQM plans. The GQM abstraction sheets can then be supplied with additional information, and updated GQM plans can later be generated from it.

During this iterative, mutual generation process, *GQMAspect* refers to the information already entered into the GQM plan, although it is not visible in an intermediate GQM abstraction sheet.

Abstraction Sheet					
GQM-Goal	Object	Purpose	Quality Focus	Viewpoint	Environment
Process	Development process	Understanding	Reliability	Software development team	Company X
Quality Focus			Variation Factors		
QUALITY FOCUS QUALITY DEFINITION Improvement Program WW Quality Model MODEL DEFINITION Q.1 What is the overall number of failures reported before delivery? MQ.1.1 Number of failure reports turned in before delivery: Q.2 What is the distribution of failures reported before delivery? MQ.2.1 For each failure reported before delivery: Report Q.3 What is the overall number of faults detected before delivery? MQ.3.1 Number of fault reports turned in before delivery: Q.4 What are the names of the top five faulty modules? MQ.4.1 For each fault detected before delivery: Name of module Q.5 What is the distribution of faults by life cycle phase? MQ.5.1 For each fault detected before delivery: Life cycle phase			VARIATION FACTORS PROCESS DEFINITION PROCESS CONFORMANCE D.1 Has the conformance of code inspections an impact on D.1.1 What is the conformance of code inspections with MD.1.1.1 For each module: Conformance of code inspections MD.1.1.2 For each module: Size of module in KLOC (K=1000) D.1.2 What is the distribution of code inspections with MD.1.2.1 For each fault detected before delivery: Life cycle phase DOMAIN CONFORMANCE D.2 Has the experience of the development team members an impact on D.2.1 How high is the experience of the development team members MD.2.1.1 For each development team member: Classification D.2.2 What is the number of faults detected before delivery? MD.2.2.1 Number of fault reports turned in before delivery		
Baseline Hypothesis			Impact on Baseline Hypothesis		
BASELINE HYPOTHESIS QUALITY DEFINITION Improvement Program WW Quality Model MODEL DEFINITION Q.1 120 failures MQ.1.1 120 Q.2 5% fatal, 15% major, 80% minor failures MQ.2.1 5%, 15%, 80% fat/maj/min Q.3 200 faults MQ.3.1 200 Q.4 Highest number expected in modules X, Y, Z MQ.4.1 X, Y, Z Q.5 10% requirements, 30% specification, 60% design MQ.5.1 10%, 30%, 60% req/spec/des			IMPACT ON BASELINE HYPOTHESIS PROCESS DEFINITION PROCESS CONFORMANCE D.1 Conformance of code inspections results in lower percent DOMAIN CONFORMANCE D.2 More experienced development team members introduce less		
FEEDBACK Ad Q.1: Programmers do not fill out failure reports regularly!					

Figure 4. The GQMaspect abstraction sheet window

Additional features of GQMaspect are:

- The ability to reuse GQM abstraction sheets, GQM plans, and parts of them
- An open file interface using an SGML-type file format
- An interface for text-processing and printing, using Framemaker and Microsoft Word.

MetriFlame – How to Use Tool Support in Measurement Management?

The use of measurement data provides a solid base for the management of improvement programmes. Based on the measurement analysis, improvements can be expressed in measurable terms, and corrective action planned during the improvement project. Unfortunately, collecting and analysing measurement data can be laborious if done manually. Proper tool support is essential to reduce the work necessary for time-consuming and ponderous measurement tasks, and also helps to reduce any resistance to measurement. The automation of measurement data collection and data analysis features enables measurement data to be used cost-efficiently and systematically.

In this section, we will describe the MetriFlame tool. It was used during the PROFES project to support measurement tasks in the application projects. During the PROFES project, MetriFlame was enhanced to support continuous assessment, and to provide reports and graphical output in html format. More information on the MetriFlame tool can be found at <http://www.ele.vtt.fi/docs/soh/metriflame/>. See also [Parviainen et al. 1997] for a more discussion on tool support for measurement programmes.

Introduction to MetriFlame

MetriFlame is a PC-based tool environment for managing measurement plans, data, and results. When MetriFlame development began at VTT Electronics in 1995, there was no tool support available that would include all measurement programme activities. There were several metric management tools like PC-Metric, Archimedes BugBase, Metricate etc., but these tools were only capable of collecting fixed metrics, and error data had to be manually entered. Furthermore, the existing commercial tools at the time did not provide any support for goal-oriented measurement programmes with variable sets of metrics.

MetriFlame is suitable for measurement data collection, metrics definition and calculation, and the presentation of analysis results in various formats. Documents and databases created during a normal software development process are typical sources of measurement data. The data collection process has been made as unobtrusive as possible, reducing any interference with the software development process to a minimum.

It is possible to automate measurement data collection and analysis with MetriFlame, and to support GQM methods where metrics may vary from

project to project. MetriFlame metrics calculation is based on the evaluation of associated formulas. Once the formulae are filled out with values and the latest data is available, the measurements can be repeated. This reduces the need for extra work each time the measurement results are calculated.

The MetriFlame tool can be used to support the following PROFES Steps:

- Step 8: Set Metrics for the Processes and Product
- Step 10: Implement and Monitor Improvements.
- Step 11: Evaluate Results

MetriFlame Tool Environment

The main elements of the MetriFlame tool environment are (see Figure 5):

1. Measurement data collection and conversion components (data sources).
2. The MetriFlame tool (data processing)
3. Display formats for metrics' results (data analysis)

The MetriFlame architecture facilitates the adding of new data sources, such as converters, and different display formats.

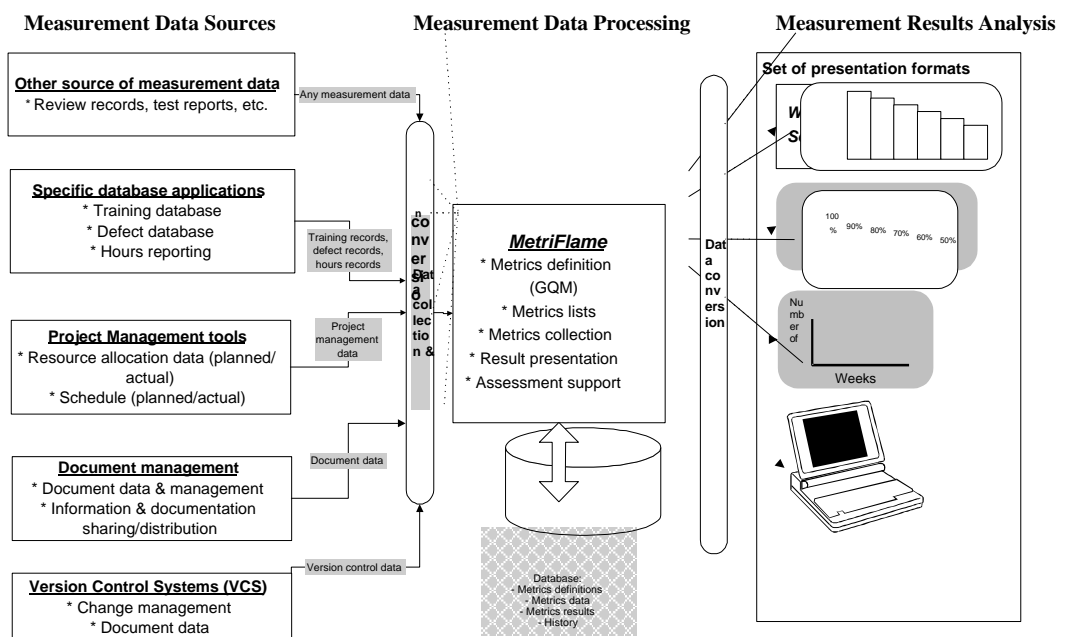


Figure 5. MetriFlame tool environment

Main Functions of MetriFlame

MetriFlame provides a solution for goal-oriented measurement programmes that collect measurement data, define and calculate metrics, and display the results in different formats for analysis sessions. MetriFlame can be connected to various data sources; the MetriFlame architecture permits both the adding of new sources of data to the system, and the viewing of the results in various formats (see Figure 5).

Briefly, MetriFlame supports:

- Importing data from data sources, such as Lotus Notes, MS Project, testing tools, defect databases, effort-tracking databases, etc.
- Managing projects and GQM plans
- Calculating various characteristics for the imported data
- Visualizing the results of calculations according to user preferences
- Following the trends of measurement results
- Automatic updating of measurement results with the latest data.

The most important MetriFlame user functions are presented in Table 2.

Table 2. Main functions of MetriFlame

1. GQM Plan-Related Management
<p><i>Managing Goals</i></p> <p style="padding-left: 40px;">Adding, editing and removing goals Goal pool</p>
<p><i>Managing Questions</i></p> <p style="padding-left: 40px;">Adding, editing and removing questions in <i>Question pool</i></p>
<p><i>Managing Metrics</i></p> <p style="padding-left: 40px;">Adding, defining, editing and removing metrics in <i>Metrics pool</i></p>
<p><i>Managing actual GQM plans</i></p> <p style="padding-left: 40px;">Adding questions and metrics to the GQM plan. Goals, questions and metrics can be selected from a corresponding pool if a suitable one already exists. The GQM plan can also be imported from GQMAspect</p> <p style="padding-left: 40px;">Linking metrics to data sources from which the data will be retrieved by MetriFlame.</p> <p style="padding-left: 40px;">Defining how to calculate the actual results of the retrieved data</p>

Managing projects

Adding, editing, and removing projects linked to GQM plans

2. Measurement Data Management

Converting data from external sources

Further processing of the data (combining fields, removing obsolete data, etc.)

Calculating

3. Measurement Results Management

Defining parameters for presentation

Viewing metrics' results according to selected output format

1. GQM Plan Management

The GQM plan can either be fed directly into MetriFlame using its own set of tools, or it can be created using *GQMaspect* and then be imported into MetriFlame. If a GQM plan is imported into MetriFlame, the following functions can be used to modify the plan when needed. It is also possible to create the GQM plan with MetriFlame; in this case these functions are used to create the GQM plan.

Management of Goals

Each GQM plan in MetriFlame is based on one or more measurement goals. MetriFlame uses a standardized presentation template for goal presentation. Measurement goals contain generic parts common to all goals, and parts that differ from goal to goal. The user defines those parts that differ from goal to goal.

Management of Questions and Metrics

The questions and metrics that will be used in the GQM plan can be selected from previously defined goals and metrics from corresponding pools in MetriFlame, or they will have to be defined for the first time. User may add, remove, and modify questions and metrics in pools and in a GQM plan.

Managing Measurement Data and Calculating the GQM Plan

When a GQM plan exists in MetriFlame, information on where the data can be retrieved and how it is calculated are defined. The MetriFlame principle is to use data already existing in different data sources within the organization. When a GQM plan exists, the data sources are linked to metrics defined in the GQM plan. The rules of data calculation are defined with the formulae.

Project Management

GQM plans stored in MetriFlame are attached to a project. The user may add, remove, or modify any projects. A single project may have multiple measurement goals and therefore multiple GQM plans.

2. Measurement Data Management

Measurement data is imported into MetriFlame from data sources via MetriFlame external converters. A separate converter is developed for each of the measurement data sources. The converters standardize the original measurement data into a common format used by MetriFlame for all its metrics processing activities. If necessary, measurement data from external sources can also be modified after conversion.

3. Measurement Results Management

Measurement results can be presented in different formats. The user can define the result graphs' parameters and attributes. If the attributes are pre-set by the user, the graph will be displayed immediately. Metrics results are available from the GQM plan at any time by simply clicking the mouse, given that the results were previously calculated. Attributes may be further modified, and the final graphs can be saved in different file formats. Output in text format allows further processing in other applications. In Figure 6, a MetriFlame presentation format example is shown.

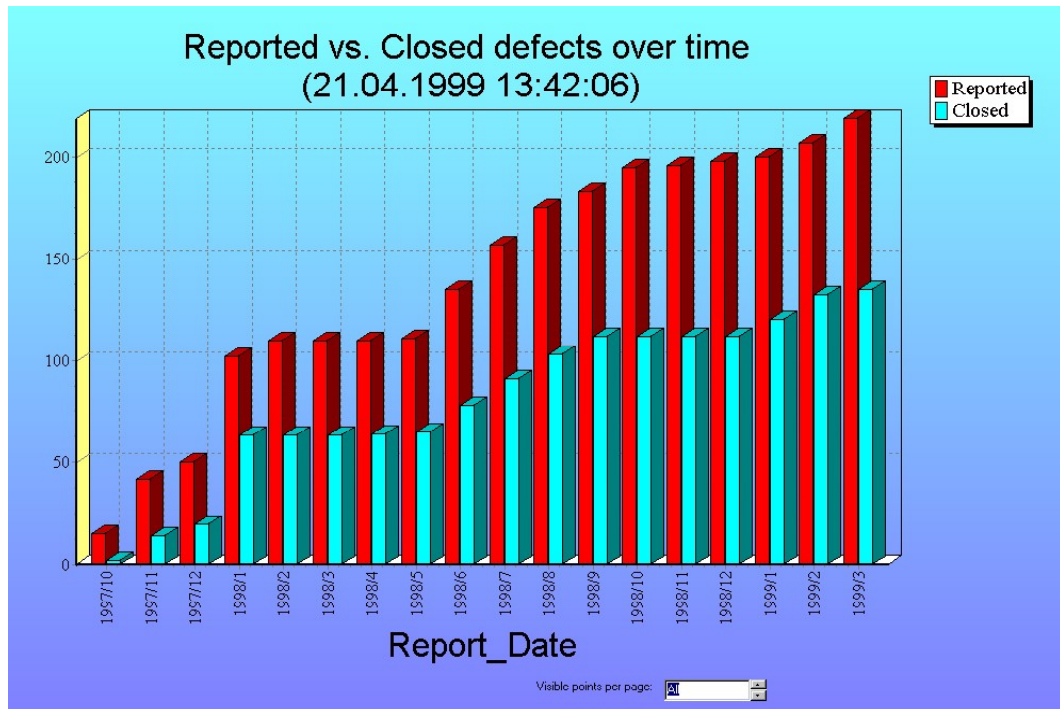


Figure 6. An example of MetriFlame result presentation

How to Monitor Process Capability During Process Improvement?

Introduction

SPICE-conformant process assessments are based on a reference process model that provides a roadmap to improve individual processes. Improvement actions resulting from a process assessment or any other origin, such as measurements, should generally result in improved process capability of the process or processes subject to changes. Only in a few cases might the current process capability level be diminished: for instance, this may happen if the organization decides to strengthen practices at lower levels and needs to focus its efforts in order to achieve this objective. Such a decision should be clearly made and documented, and should not simply be a by-product of process changes addressing other objectives and processes. Once process changes have been completed, we recommend repeating a process assessment at least for those processes affected by the process changes. In addition to two subsequent process assessments, continuous assessment techniques help the monitoring process capability. A process capability trend analysis will make the effect of process changes on process capability visible over a given period of time.

Once improvement actions are completed and improvements achieved, new priorities are set for the organization and the projects. At this stage, improvement results have to be sustained to ensure that they will last. Repeating process assessment when improvements are completed and using continuous assessment techniques afterwards will ensure that attention is focused on the newly achieved capability. Again, a process capability trend analysis will facilitate this effort.

Process assessment data are available in process assessment tools such as Bootsampl, so that they can be easily used to produce capability trend graphs. A tool supporting the production of such graphs is desirable when building a PROFES tool environment. The PROFES project has developed an example of such a tool, which is briefly presented in the following section.

Introduction to the PROFES Capability Trend Analysis Tool

Figure 7 shows the main functionality of the PROFES capability trend analysis tool.

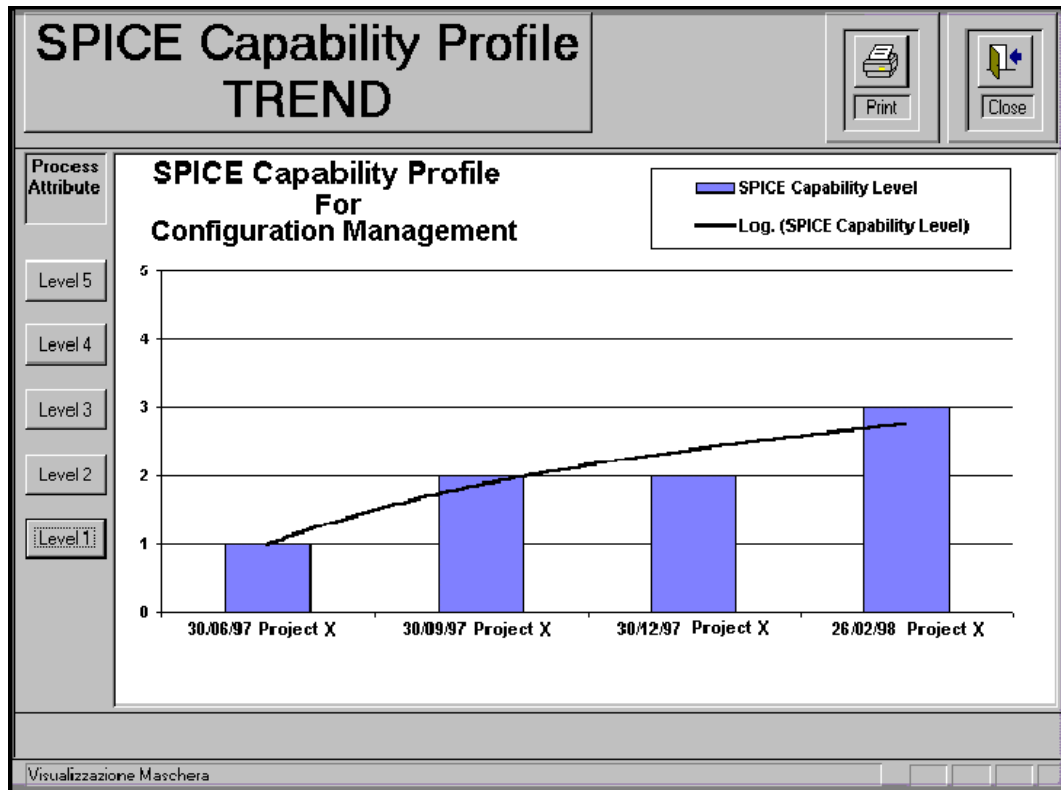


Figure 7. PROFES capability trend analysis tool

The diagram demonstrates the process capability trend for a defined process. Process capability was evaluated in subsequent process assessments in the same project, as shown in the figure. The graph makes the positive trend clearly visible, with a rapid improvement from Level 1 to Level 2, with a slower transition from Level 2 to Level 3.

Figure 8 shows the same trend graph using the BOOTSTRAP quartiles:

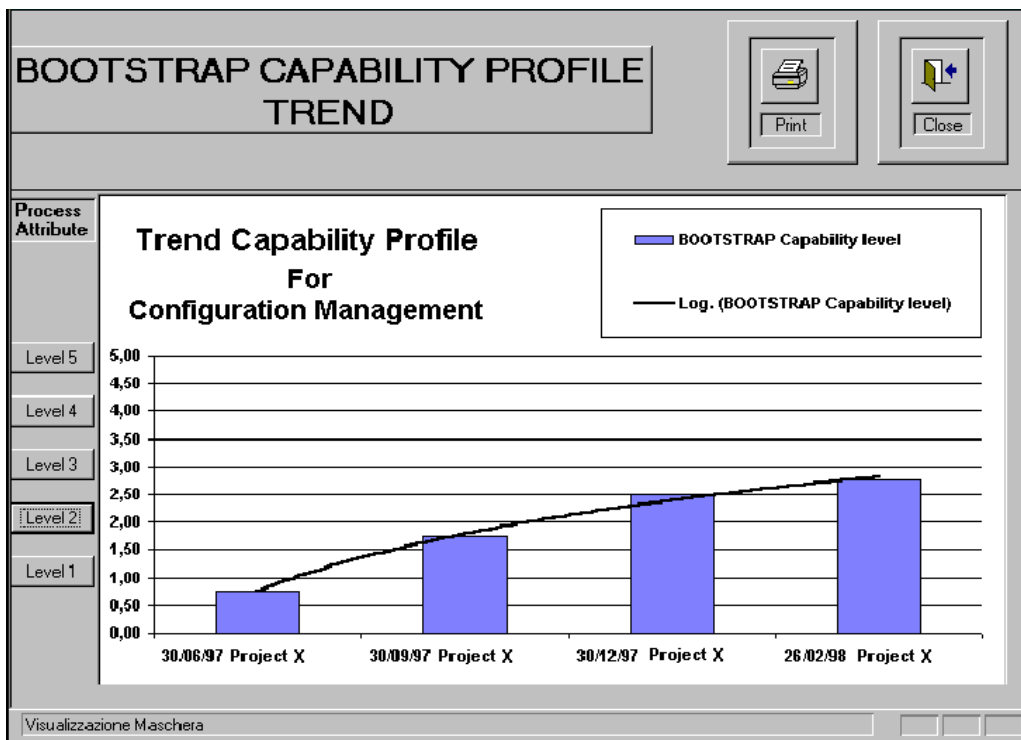


Figure 8. Capability trend using BOOTSTRAP quartiles

In this case, better granularity is provided that allows a more accurate review of what happened in between the subsequent assessments. The capability level does not move directly from one level to the next, but a smoother transition occurs.

For both a pure SPICE profile and a BOOTSTRAP one, assessors can demonstrate the process attribute trend, if they wish to get a better insight into the capability levels. Figure 9 zooms into Level 1 process attributes, which show that practices for Level 1 were not fully implemented during the first assessment.

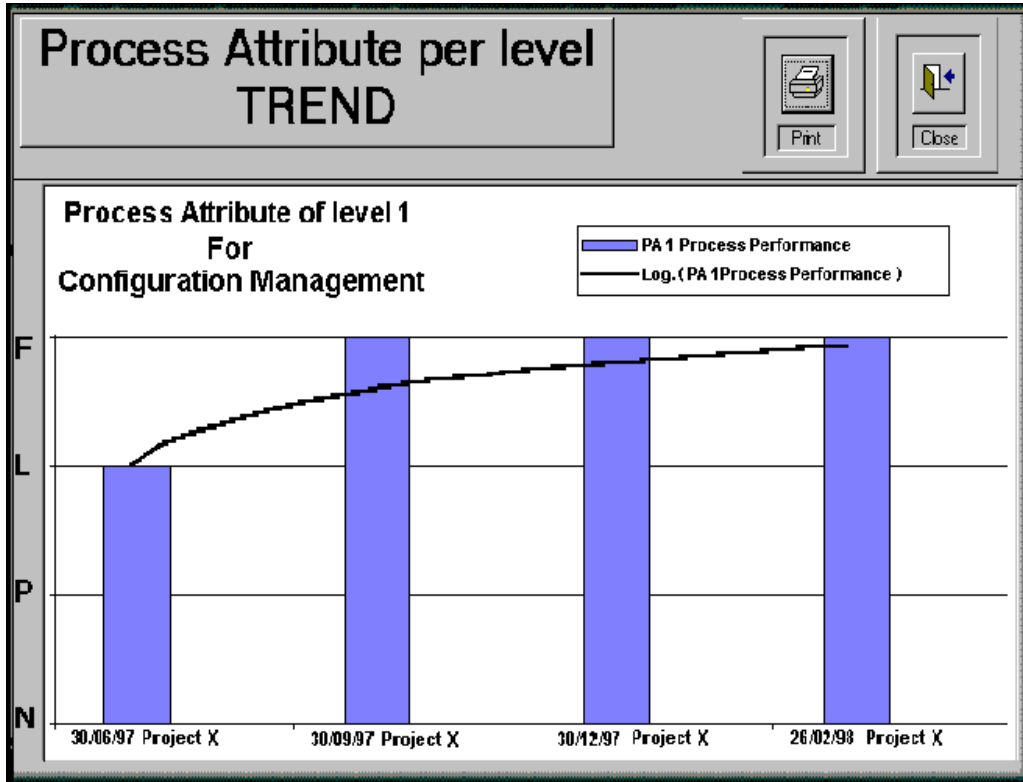


Figure 9. Capability trend for Level 1 process attribute

It might then be desirable to check the situation with Level 2 process attributes, as shown in Figure 10.

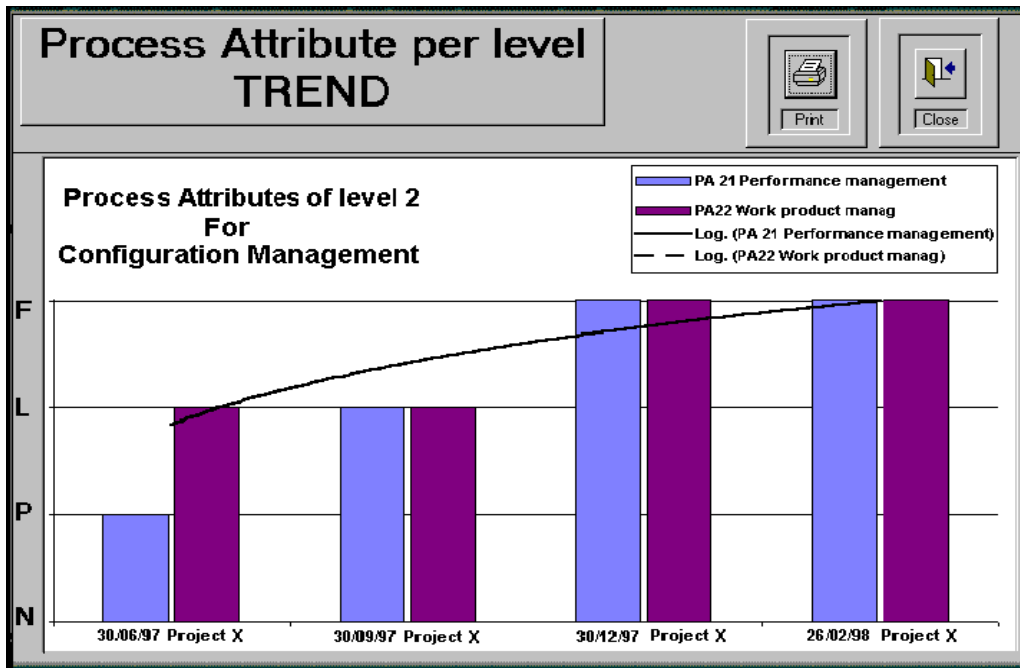


Figure 10. Capability trend for Level 2 process attributes

Figure 10 shows how practices at Level 2 were gradually implemented, providing a better insight into the improvement trends.

Quite often, improvement addresses not just one process but a set of related processes together. If this is the case, it is interesting to monitor the capability trend of all the processes affected by improvements. Figure 11 shows how the PROFES capability trend analysis tool can display the capability trend of multiple processes on the same screen.

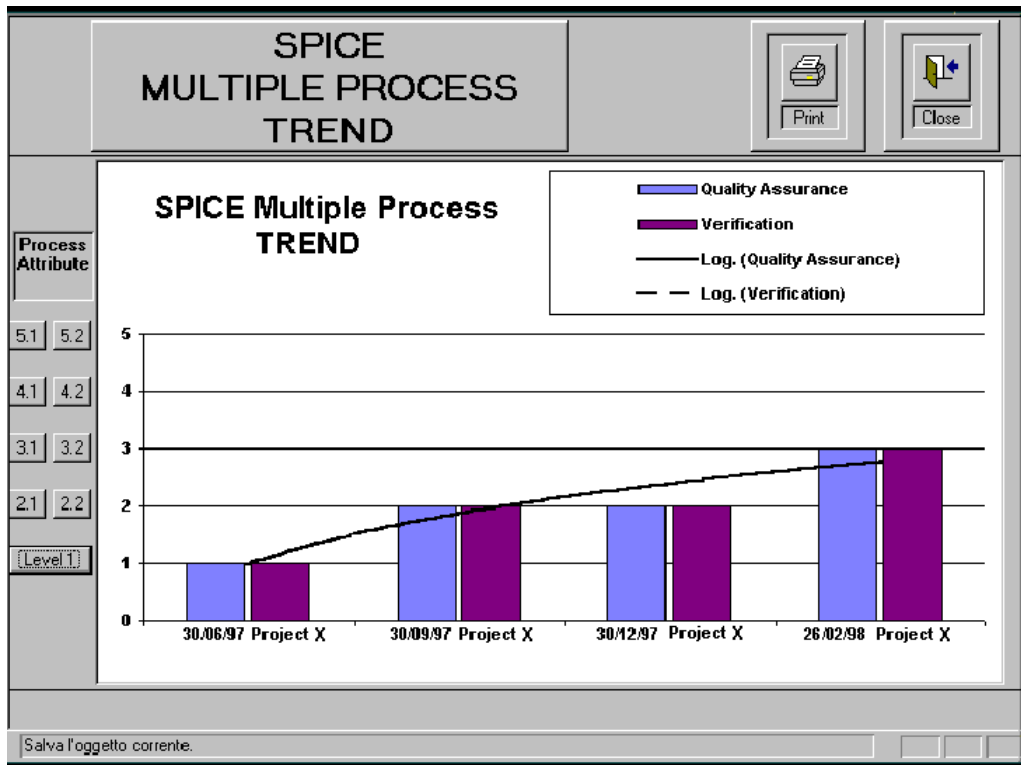


Figure 11. Capability trend analysis of multiple processes

Again, the trend analysis of selected process attributes can be demonstrated, if better insight is needed.

References to Appendix 5

[Basili et al. 1994b] Victor R. Basili, Gianluigi Caldiera, and H. Dieter Rombach. "Goal Question Metric Paradigm". In John J. Marciniak, editor, *Encyclopaedia of Software Engineering*, volume 1, 1994, pages 528–532. John Wiley & Sons.

[Birk et al. 1998] Birk, A., Kempkens, R., Rombach, H.D., Ruhe, G., "Systematic Improvement of Software Engineering Processes", *Proceedings of Fruehjahrstagung Wirtschaftsinformatik '98*, Braunschweig: Vieweg, 1998, pp. 265-280.

[Briand et al. 1996] Lionel Briand, Christiane Differding, and H. Dieter Rombach: "Practical guidelines for measurement-based process improvement". *Software Process Improvement & Practice*, December 1996, 2(4):253-280.

[Greese et al. 1995] Christiane Greese, Barbara Hoisl, and Jürgen Wüst: "A Process Model for Planning GQM-based Measurement". Technical Report, STTI-95-04-E, Software Technology Transfer Initiative (STTI), University of Kaiserslautern, October 1995.

[Hoffman et al. 1996] Hoffmann, M., Birk, A., van Els, Kempkens, R., "GQMaspect User Manual V1.0", Fraunhofer IESE, November 1996.

[Latum et al. 1998] Frank van Latum, Rini van Solingen, Markku Oivo, Barbara Hoisl, Dieter Rombach, and Günther Ruhe. "Adopting GQM-Based Measurement in an Industrial Environment". *IEEE Software*, 15(1), January 1998, pp. 78–86.

[Parviainen et al. 1997] Parviainen, P., Järvinen, J. and Sandelin, T. "Practical Experiences of Tool Support in a GQM-based Measurement Programme", *Software Quality Journal*, Vol. 6, No.4, December 1997, pp. 283-294.



APPENDIX 6

THE PROFES GLOSSARY

This appendix defines the main terms associated with the PROFES improvement methodology. The intention of this glossary is to describe those terms that are relevant and specific to the PROFES improvement methodology. The document is structured according to the PROFES elements.

Definitions

For the purposes of PROFES improvement methodology, the definitions given in ISO 8402, ISO/IEC 2382-1, ISO/IEC 2382-20, ISO/IEC 9126, ISO/IEC 12207, ISO/IEC15498, ISO/IEC 15504 and ISO/IES 15939 apply together with the following definitions. Some common dictionary terms have been included to clarify the specific meaning in the context of the PROFES improvement methodology.

GENERAL PROFES TERMINOLOGY

Baseline

A baseline is a quantitative or qualitative description of the current situation. Within an improvement programme, the baseline is used as a basis for goal definition and to evaluate changes that were implemented to improve the situation.

Improvement

An improvement is a change for the better when compared to the old situation (baseline), according to predefined criteria.

Goal

There are different types of goals that can be applied on different organizational levels, i.e. SPU and project level:

- Business goals
- Improvement goals: can be process-related, product-related, organization-related, project-related, etc.
- Measurement goals
- GQM goals

Please note that purpose and content is highly dependent on the type of the goal in question, i.e. different types of goals are not interchangeable.

Improvement Goal

The target for the outcome of improvement activities. There are different types of improvement goals:

- Organizational improvement goals
- Project-related improvement goals
- Process-related improvement goals

- Product-related improvement goals

Model

A representation of an artifact or activity intended to explain the behaviour of some aspect of it. The model is less complex or complete than the activity or artifact themselves. A model is considered to be an abstraction of reality.

Empirical Model

An empirical model is a set of propositions or equations describing in simplified form some aspects of real-life experience.

Process Model (PM)

A (software) process model is a model of an actual or hypothetical (software) process. A (software) process can be considered to be a specific type of (software-related) know-how. Conceptually, a process consists of a set of interrelated entities, such as activities, artifacts, and agents. A process model can also represent the behaviour of entities and relationships.

A process model can be both descriptive and prescriptive.

Descriptive Process Model (DPM) – a process model is descriptive if it represents the actual process.

Prescriptive Process Model (PPM) – a process model is prescriptive if it is used for specifying a process that should be followed.

Product

There is a distinction between work products and end products. A work product is used within the development organization, e.g. requirement definition, design documents, test case descriptions, and inspection reports. Work products are necessary for developing and maintaining the end product. End products are those products that are delivered to the customer. Please note that some work products may also be or will become end products, for example user manuals or requirement specifications.

If used alone, the word product refers to the set of end products that collectively form the final product, such as a mobile phone, a medical instrument or a petrol dispenser.

GQM TERMINOLOGY

Goal/Question/ Metric (GQM) Approach

The GQM (Goal/Question/Metric) is an approach for goal-oriented measurement in software projects. It consists of three components:

- GQM paradigm
- GQM plans or GQM models
- GQM method

The GQM paradigm includes the basic idea that measurement should be goal-driven, as well as several principles to be followed when applying GQM-based measurement.

A GQM plan or GQM model documents the refinement of a precisely specified measurement goal by a set of questions into a set of metrics. Thus, a GQM plan documents which metrics are to be used to achieve a measurement goal and why – the questions provide the rationale underlying the selection of the metrics. On the other hand, the GQM plan is used to assist analysis tasks, as it documents purpose for which the respective data were collected. Please note that the terms “GQM plan” and “GQM model” are used as synonyms in literature; we will use the term GQM plan here.

The GQM method provides assistance on how to set up and perform GQM-based measurement programs.

Feedback Session

A Feedback Session is a meeting that is regularly held during a measurement programme. The purpose of the meeting is to present collected measurement data to the project team and management, and analysis results derived from that data. The project team and management then have the opportunity to interpret the analysis results, and to draw conclusions that may lead to improvement suggestions for product development and measurement.

GQM Plan

A GQM plan documents the refinement of a precisely specified measurement goal by a set of questions into a set of metrics. Thus, a GQM plan documents which metrics are to be used to achieve a measurement goal and why – the questions provide the rationale underlying the selection of the metrics. On the other hand, the GQM plan is used to assist analysis tasks, as it documents for which purpose the respective data was collected. Please note that the terms “GQM plan” and “GQM model” are used as synonyms in literature; we will use the term GQM plan here.

Measure

Noun: the number or category assigned to an attribute of an entity by making a measurement.

Verb: to make a measurement.

Measurement

The use of metrics to assign a value from the measurement scale to an attribute or entity.

Metrics

The defined measurement method and the measurement scale.

Measurement Plan

The measurement plan is an application of the GQM Plan. It defines:

- Who collects (role) data?
- What kind of data (metrics) is collected?
- When is the data collected?
- How is the data collected?

GQM Goal (also known as Measurement Goal)

A GQM (measurement) goal is a formal measurement goal definition specifying the following measurement goal dimensions:

- Object: Which entity is the subject of measurement?
- Focus: Which aspects (attributes) of the object are of interest?
- Purpose: For what purpose is the measurement made?
- Viewpoint: To whom are the measurement results of interest?
- Context: In which organizational environment is the measurement programme conducted?

Abstraction Sheet

The Abstraction Sheet is used to support the elicitation of knowledge and experience necessary to construct a GQM plan. The Abstraction Sheet is a template consisting of a header section that refers to the GQM goal definition, and a content section that contains the following four information fields:

- Quality focus
- Baseline hypothesis
- Variation factors
- Impact on baseline hypothesis

The understanding, knowledge, experience, and expectations collected in the content section are usually gathered during interviews.

Product-Process Dependency (PPD) TERMINOLOGY

Product-Process Dependency (PPD)

A product-process dependency (PPD) is a causal relationship between aspects of a software development process and aspects of a software product that is developed by applying that same process. It is assumed that a software process affects and determines the developed product and its qualities.

Please note that in the context of PPDs, process denotes a process as used in ISO 15504 terminology, or the entire software development for which a software development practice is applied, in order to yield specific software quality. According to ISO 15504 terminology, only the processes defined in the Bootstrap methodology are used. Depending on the particular context in which the PPD-related term process is used, software development task can also be used.

N.B. In the context of PPDs, practice denotes a software engineering or management activity (cf. the term “practice” in ISO 15504 terminology). This understanding of practice involves the use of resources, a set of such activities (cf. the term “process” in ISO 15504 terminology), a standard, a policy, a technique, a method, or a tool that is applied in a specific software development task in order to yield specific software quality. In most cases, it is widely synonymous with the term “practice” in ISO 15504. Depending on the particular context in which the PPD-related term “practice” is used, software development practice, process, or software engineering technology can also be used.

PPD Model

A PPD model is an abstraction of a PPD. There are different types of PPD models. Each PPD model type has a specific perspective on the described PPD. In addition, PPD models can differ in their level of detail and the relational semantics of their effect.

Different PPD model types have different application areas for which they are particularly well suited.

In PROFES, we distinguish three basic PPD model types:

- Process impact model
- Technological impact model
- Contextual impact model

These PPD model types describe PPDs at increasing levels of detail. Process impact PPD models are the least detailed ones. Each PPD model type is described as follows.

Process Impact PPD Model

A process impact PPD model describes the impact of a software engineering process on product quality. Its effect can be shown as follows:

Process --> Product Quality

We define the software engineering processes using the ISO 15504 process dimension. Product quality is described in accordance with ISO 9126 quality factors, development cost, and development time.

Technological Impact PPD Model

A technological impact PPD model describes the impact of software engineering technology on product quality, when applied to a specific software engineering process. Its effect can be shown as follows:

Technology --> Process --> Product Quality

A software engineering technology is any technique, method, tool, or artifact which is established or used within a software engineering, support, or management activity, and which contributes to the creation of a software product.

Contextual Impact PPD Model

A contextual impact PPD model describes the impact of the application context of a software engineering technology on its effectiveness. The technology is applied to a specific software engineering process. Effectiveness is determined in terms of meeting certain product quality criteria. Its effect can be shown as follows:

Context --> Technology --> Process --> Product Quality

In the PROFES improvement methodology the application context of a software engineering technology is described in terms of context characteristics. A context characteristic can be any aspect of a software project and its organizational environment, such as project size, development approach, process capability, organizational culture, personnel characteristics, and product requirements.

Product Quality

In the context of PPDs, product quality is the total sum of features and characteristics of a product or services that affect its ability to satisfy stated or implied needs.

In the context of PPDs, the product is the final software product that is then embedded into the entire product.

For PPDs, the set of product qualities is limited to those defined by ISO 9126, i.e. reliability, maintainability, functionality, usability, efficiency, portability, and their sub-aspects, plus the qualities cost and time-to-market.

Context

When context is related to PPDs, it means a set of procedural, organizational, personal, social, historical, and infrastructure characteristics of a software development organization or unit. It describes whether the application of a practice to a process contributes significantly to the achievement of product quality.

Context is defined in terms of attribute/value pairs. Attributes are also called contextual factors. Values are also called contextual characteristics.

A contextual definition is always specific to a particular environment, such as an individual software organization or a branch of software engineering, e.g. embedded systems development.

EXPERIENCE FACTORY TERMINOLOGY

Experience Factory

The Experience Factory is a logical and/or physical organization that supports project developments by analysing and synthesizing all kinds of experience, acting as a repository for such experience, and supplying that experience to various projects on demand.

Experience Package

The experience package is the main product of the Experience Factory. It consists of a central element, such as an experience model. This might be a life cycle product or process, a mathematical relationship, an empirical or theoretical model, or a data base, which is packaged together with specific context information in a specific form that allows easy access and reuse when required. The purpose of the experience package is to provide relevant and easy-to-use experience to appropriate software projects on demand.

Experience Base

The experience base is a set of integrated experience packages. They can be represented in arbitrary form, e.g. as handbooks, reports, presentation slides, tools, HTML pages, or database. The experience base is an essential part of the Experience Factory.

ASSESSMENT TERMINOLOGY

Process

A set of inter-related activities, which transform input into output.

(N.B. The term “activities” includes use of resources).

Software Process

The process or set of processes used by an organization or project to plan, manage, implement, monitor, control, and improve its software-related activities.

Practice

A software engineering or management activity that contributes to the creation of output (work products) in a process or enhances the capability of a process.

Base Practice

A software engineering or management activity that addresses the purpose of a particular process.

Management Practice

A management activity or task that addresses the implementation or institutionalization of a specified process attribute.

Organizational Unit

A part of the organization that is the subject of an assessment. An organizational unit deploys one or more processes that have coherent process context and operate within a coherent set of business goals and practice.

N.B. An organizational unit is typically part of a larger organization, although in a small organization, the organizational unit may refer to the whole organization. For example, an organizational unit might be:

- A specific project or set of (related) projects
- A unit within an organization focused on a specific lifecycle phase (or phases), such as acquisition, development, maintenance, or support
- A part of an organization responsible for all aspects of a particular product or product set.

Process Capability

The ability of a process to achieve a required goal.

Process Capability Level

A point on a six-level ordinal scale of process capability that represents the increasing capability of the performed process. Each level builds on the capability of the level below it.

Process Purpose

The high-level, measurable objectives of performing the process, and the likely outcome of effective implementation of the process.

Process Attribute

A measurable characteristic of process capability applicable to any process.

Defined Process.

The operational definition of a set of activities for achieving a specific purpose. A defined process may be characterised by standards, procedures, training, tools and methods

Process Improvement

Action taken to change an organization's processes in order to meet organizational business requirements and achieve business goals more effectively.

Process Improvement Action

An action planned and implemented to improve part or all of the software process. A process improvement action can contribute to the achievement of more than one process goal.

Process Outcome

An observable result of the successful implementation of a process.

Objective Evidence

Qualitative or quantitative information, records, or statements of fact referring to the characteristics of an item or services, or to the existence and implementation of a process element, which is based on observation, measurement, or test result, and which can be verified (ISO 10011:1994).

Measurement-Based Continuous Assessment (MCA)

An approach to performing frequent assessments based on defining a distinct set of assessment indicators that are integrated into a measurement plan. MCA is a special variant of continuous assessment.

Assessment Indicator

An objective attribute or characteristic of a practice or work product that supports judgement of the performance or capability of an implemented process.

Continuous Assessment

An approach to performing frequent assessments, based on defining a distinct set of assessment indicators that can be checked during project implementation.

Index to Appendix 6: The PROFES Glossary

Abstraction Sheet.....	6
Assessment Indicator.....	13
Base Practice.....	11
Baseline.....	2
Context.....	9
Contextual Impact PPD Model.....	8
Continuous Assessment.....	13
Defined Process.....	12
Empirical Model.....	3
Experience Base.....	10
Experience Factory.....	10
Experience Package.....	10
Feedback Session.....	4
Goal.....	2
Goal/Question/ Metric (GQM) Approach.....	4
GQM Goal (also known as Measurement Goal).....	6
GQM Plan.....	5
Improvement.....	2
Improvement Goal.....	2
Management Practice.....	11
Measure.....	5
Measurement.....	5
Measurement Plan.....	5
Measurement-Based Continuous Assessment (MCA).....	13
Metrics.....	5
Model.....	3
Objective Evidence.....	13
Organizational Unit.....	11
PPD Model.....	7
Practice.....	11
Process.....	10
Process Attribute.....	12
Process Capability.....	12
Process Capability Level.....	12
Process Impact PPD Model.....	8
Process Improvement.....	12
Process Improvement Action.....	12
Process Model (PM).....	3
Process Outcome.....	13
Process Purpose.....	12
Product.....	3
Product Quality.....	9
Product-Process Dependency (PPD).....	7
Software Process.....	11
Technological Impact PPD Model.....	8

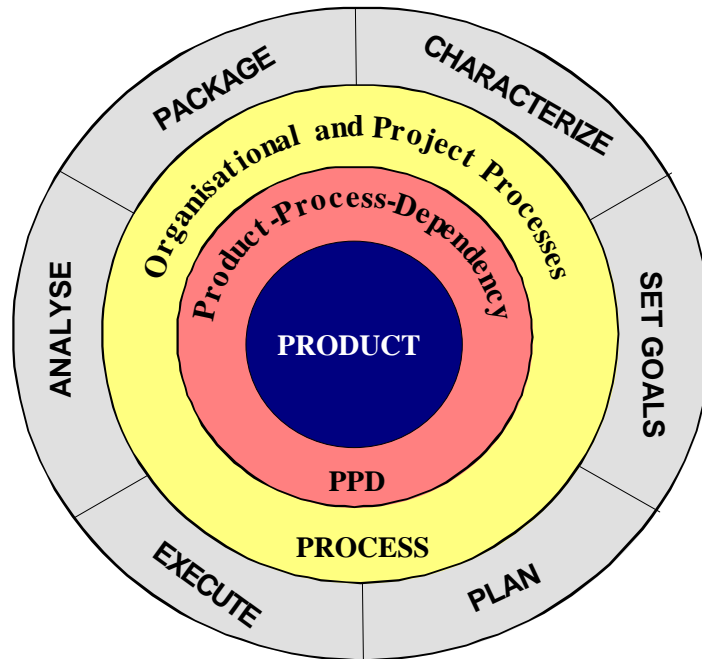


APPENDIX 7

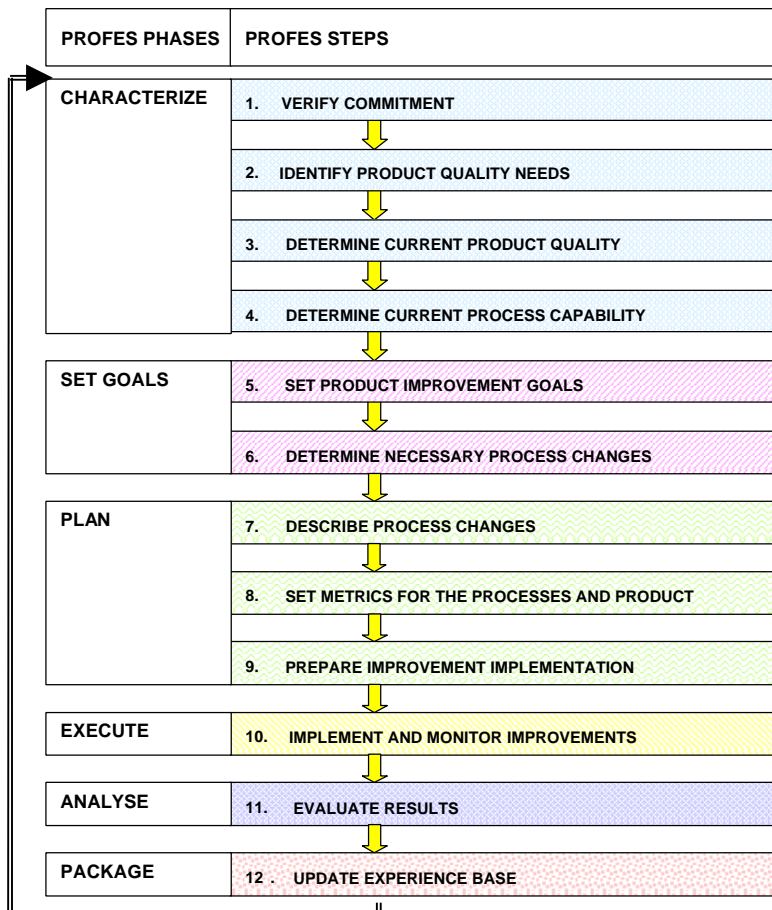
PROFES QUICK REFERENCE CHART

This appendix presents the PROFES improvement methodology in a concise reference chart to be used for quick check on the PROFES phases and steps. Also step goals, activities and major input and output work products are listed.

QUICK REFERENCE CHART



The PROFES improvement cycle



Phases and steps of the PROFES improvement methodology



QUICK REFERENCE CHART

Step 1 Verify commitment	
Goals	Activities
<ul style="list-style-type: none"> The organization's business needs and improvement objectives for product and process quality are identified. Product quality characteristics, ongoing improvement initiatives, and their priorities are identified. Commitment of top and middle management is verified. Commitment of project members is verified. Contextual info of the organization & projects is defined. An overall plan for improvement activities is defined. 	<ul style="list-style-type: none"> Identify the organization's business needs and improvement objectives Motivate top and middle management Motivate project members Define organizational context Define overall plan and schedule
Input	Output
Organizational level: <ul style="list-style-type: none"> General organizational information Business goals Customer survey results Market research results Customer feedback Organizational context information Project level: <ul style="list-style-type: none"> Project environment specifics Product development goals 	Organizational level: <ul style="list-style-type: none"> Commitment of top and middle management Preliminary product and process improvement needs Organization's classification Overall improvement plan Project level: <ul style="list-style-type: none"> Commitment of project management and members Project classification Overall improvement plan

Step 2 Identify product quality needs	
Goals	Activities
<ul style="list-style-type: none"> Product quality needs are known and presented in the form of a product quality profile Preliminary product quality goals are set 	<ul style="list-style-type: none"> Survey product quality needs Document product quality needs Set preliminary product quality goals
Input	Output
<ul style="list-style-type: none"> Customer survey results Market research results Customer feedback Business goals ISO9126 Preliminary product quality needs 	<ul style="list-style-type: none"> Product quality needs Product quality profile Preliminary product quality goals

Step 3 Determine current product quality	
Goals	Activities
<ul style="list-style-type: none"> Determine current status of product quality 	<ul style="list-style-type: none"> Acquire product quality data Evaluate current status of product quality
Input	Output
<ul style="list-style-type: none"> Application domain characteristics Measurement data ISO9126 Product quality profile Experience base 	<ul style="list-style-type: none"> Current status of product quality

Step 4 Determine current process capability	
Goals	Activities
<ul style="list-style-type: none"> Current process capability is determined Process improvement recommendations are documented and communicated 	<ul style="list-style-type: none"> Preparation Execution Reporting
Input	Output
<ul style="list-style-type: none"> Business goals Process descriptions Quality manuals Organizational characteristics Project plans Design documents Measurement data 	<ul style="list-style-type: none"> Process capability profiles Process assessment report and profiles Descriptive process models Preliminary improvement plan

Step 5 Set product improvement goals	
Goals	Activities
<ul style="list-style-type: none"> Set Product improvement goals 	<ul style="list-style-type: none"> Analyse product quality discrepancies Identify product improvement areas Prioritize product improvement areas Set the product improvement goals
Input	Output
Business goals Product quality needs <ul style="list-style-type: none"> Product quality target profile Current status of product quality Process assessment reports and profiles Descriptive process models Preliminary product quality goals Product characteristics 	<ul style="list-style-type: none"> Product improvement goals

Step 6 Determine necessary process changes	
Goals	Activities
<ul style="list-style-type: none"> Identify and select process changes necessary to achieve the product improvement goals. Document the decisions on necessary process changes for later evaluation of the improvement programme 	<ul style="list-style-type: none"> Identify product quality goal Identify processes to be improved Retrieve relevant PPD models Construct characterization questionnaire Characterize the project Rank PPD models Select improvement actions
Input	Output
<ul style="list-style-type: none"> Product improvement goals Process assessment reports and profiles (from Step 4) PPD repository Preliminary improvement plan (from Step 4) 	<ul style="list-style-type: none"> Process changes to be implemented in the improvement programme Characterization of the forthcoming project or improvement programme

Step 7 Describe process changes	
Goals	Activities
<ul style="list-style-type: none"> Agree and document prescriptive process model Achieve clear understanding of the processes in order to define the metrics in the following step 	<ul style="list-style-type: none"> Mark processes/practices in the current process model, which have to be changed Develop prescriptive process model Communicate prescriptive model to process participants
Input	Output
<ul style="list-style-type: none"> Descriptive process model (from Step 4) Selected list of process changes (from Step 6) 	<ul style="list-style-type: none"> Prescriptive process model (including selected process changes) Training/presentation material for the new process

Step 8 Set metrics for the processes and product improvements	
Goals	Activities
<ul style="list-style-type: none"> Define questions and metrics related to the product quality goals Define questions and metrics related to the process performance goals Define questions and metrics related to the product-process dependency goals Construct GQM plan and measurement plan 	<ul style="list-style-type: none"> Define measurement goals Conduct GQM interviews Define questions and hypotheses Define and check metrics Produce GQM plan and measurement plan
Input	Output
<ul style="list-style-type: none"> Prescriptive process model (including selected process changes) Product quality and target profile (from Step 2) Current status of product quality (from Step 3) Product improvement goals (from Step 5) Process assessment reports and profiles (from Step 4) PPD models (from Step 6) 	<ul style="list-style-type: none"> GQM abstraction sheets GQM plan Measurement plan

Step 9 Prepare improvement implementation	
Goals	Activities
<ul style="list-style-type: none"> Plan process changes and allocate sufficient resources to implement them Plan improvement progress meetings 	<ul style="list-style-type: none"> Plan process improvement progress meetings Make time planning and resource allocation Kick-off process changes
Input	Output
<ul style="list-style-type: none"> Development project plan Preliminary improvement plan (from Step 4) Selected list of process changes (from Step 6) Prescriptive process model (from Step 7) GQM deliverables (from Step 8) 	<ul style="list-style-type: none"> Process improvement action plan On-line process support

Step 10 Implement and monitor improvements in the development project	
Goals	Activities
<ul style="list-style-type: none"> Implement selected process changes according to process improvement plan Collect data and prepare measurement results for each feedback session Hold feedback sessions 	<ul style="list-style-type: none"> Implement process changes Collect measurement data Prepare and select measurement data Perform GQM feedback sessions
Input	Output
<ul style="list-style-type: none"> Prescriptive process model GQM plan Measurement plan Process improvement plan Development project plan 	<ul style="list-style-type: none"> Measurement data Feedback session report(s) with visualized measurement data Description of corrective actions taken Prescriptive process model applied in practice

Step 11 Evaluate Results	
Goals	Activities
<ul style="list-style-type: none"> Evaluate effect of the improvement programme on final product quality Evaluate changes to the software engineering process, methods, and tools Gather and evaluate "lessons learned" Support, modify, or reject used PPD models 	<ul style="list-style-type: none"> Evaluate the measurement results Support, modify, or reject used PPD models
Input	Output
<ul style="list-style-type: none"> PPD models Prescriptive process model Abstraction sheets GQM plan GQM measurement plan Measurement data Feedback session reports 	<ul style="list-style-type: none"> Preliminary experience packages Evaluated PPD models

Step 12 Update Experience Base	
Goals	Activities
<ul style="list-style-type: none"> Package and store all information gained during the project in the experience base for future reuse 	<ul style="list-style-type: none"> Package information Store relevant information in the experience base
Input	Output
<ul style="list-style-type: none"> Evaluated PPD models Experience base Process models GQM plan Feedback session reports 	<ul style="list-style-type: none"> Updated experience base with generalized: <ul style="list-style-type: none"> PPD models Process models GQM plans